Model Drift Detection Using Dynamic Thresholding

Daqian Zuo
daqianz2@illinois.edu

Derek Yang
mcy3@illinois.edu

Tianyang Liao
tl65@illinois.edu

Sizheng Zhang
sizheng3@illinois.edu

ABSTRACT

Recent advancements in machine learning and artificial intelligence have fueled interest in scalable data analysis systems. However, a critical challenge arises in the form of drift in real-world data, where models trained on historical data may face difficulties adapting to unforeseen patterns or distribution shifts. This drift poses a significant obstacle to the reliability of data analytics models in dynamic environments, emphasizing the need for adaptability to changing conditions.

This paper explores an adaptive approach for detecting model drift in deep-learning systems deployed in dynamic environments. Rather than relying on static thresholds, we propose a method that dynamically adjusts to evolving data distributions. Leveraging key metrics such as training probability and information entropy, our approach aims to provide a more responsive solution to the challenges posed by changing environmental conditions. Through experimentation across various deep-learning architectures, we assess the effectiveness of our adaptive drift detection method in comparison to existing approaches. We observed a maximum of 90% increase in the detection precision compared to traditional methods in some specific datasets. This result shed light on the potential of adaptive techniques for addressing the inherent complexities associated with model drift in dynamic deployment scenarios.

1 INTRODUCTION

As visual data modalities, including images and videos, proliferate in today's digital system, there has been a growing imperative for specialized Database Management Systems (DBMS) capable of effectively analyzing these high-dimensional datasets [13]. Although contemporary systems demonstrate commendable efficiency in query execution, they are notably vulnerable to shifts in input data distribution, which is known as concept drift. This susceptibility becomes particularly obvious in circumstances where the input video data is processed as continuous streams. [8]

A preliminary step in addressing concept drift involves its timely detection and identification within these data streams. Extant literature on concept drift detection has explored methodologies such as Discriminative Reconstruction Autoencoders (DRAE) [25] and Local Outlier factor (LOF) [3], but these techniques were engineered to handle model drifts in low-dimensional data and thus exhibit sub-optimal performances when applied to high-dimensional data, such as videos.

To ameliorate the limitations inherent in detecting concept drifts in complicated, high-dimensional data streams, researchers have refined their algorithms and detection frameworks. For instance, ODIN [22] introduced DA-GAN, a combination of Adversarial Autoencoder (AAE) and generative Adversarial Network (GAN) to detect changes in data distribution by ensuring minimal information loss during encoding. Other distribution-based concept drift detection systems include ADWIN [2] and VFDTc [9], which compare two data windows to determine whether a change in data is introduced or not [10].

In traditional classification tasks using neural networks, the softmax function conventionally serves as the final layer, transforming raw logits into interpretable probabilities. Typically, the classification decision is derived from the argmax of the softmax scores. To address the challenge of model drift, our focus shifts to the observation that high-accuracy models tend to yield markedly elevated softmax scores for the correct class, dwarfing scores for other classes. Consequently, when exposed to out-of-domain data, the model should ideally exhibit a diminished argmax score, indicating a lack of confidence in its prediction. With that in mind, setting up threshold to filter out low confident samples as drifted items, and treat high confidence sample as in-domain data, we should be able to achieve an ideal result on both drift detection for out-of-domain samples and classification on in-domian samples.

In this paper, we introduce the dynamic threshold method, a novel approach designed to enhance the detection of model drift in neural networks. Central to this method is the innovative use of key metrics—training probability and information entropy—which are dynamically adapted to detect shifts in model performance and data distribution. We chose various metrics that measure the certainty and confidence in a model's predictions, providing a quantifiable way to assess the predictability and stability of a model's output.

Within this particular context, we consider adopting the transformer architecture to be an compelling and well-suited pursuit, considering that transformers inherently possess an encoder-decoder structure with its multi-head self-attention mechanism [24]. The challenge would be to train the model to encapsulate enough information to differentiate various features in order to later detect potential model drifting. It is worth noting that there have been notable instances of utilizing vision transformers[7] as an image encoder for tasks such as image classification [26], object detection [15], and semantic segmentation [23].

We present an assessment regarding the efficacy of the combination between dynamic thresholding and neural networks such as Convolutional Neural Networks (CNNs) based concept drift models and transformer-based concept drift models. Although there are some classical techniques, such as LOF, addressing the model drift of low-dimensional data, they usually do not perform well when it comes to high-dimensional data. The exploration of evaluation of high-dimensional such as images will be an intriguing direction. Our investigation primarily revolves around two pivotal aspects: Firstly, we will build CNNs that are suitable to the provided dataset and integrate the dynamic thresholding to implement concept drift

detection. Secondly, we will explore whether the Transformer-based model's capabilities in outlier detection outperform the CNN-based models across different datasets. The evaluation section of this paper will provide an analysis of Transformer-based classification and drift detection in comparison to the conventional CNN approach alongside popular methods like LOF and AAE across different proportions of outliers. By employing diverse datasets, such as MNIST and CIFAR-10, our study aims to elucidate whether the incorporation of Transformers and dymanic thresholding method yields enhanced accuracy compared to the baselines, thereby shedding light on the transformative potential of these models in the realm of image classification and drift detection for in-domain and out-of-domain data. More specifically, we achieved different levels of increase in accuracy compared to traditional outlier detection models based on statistical methods across different datasets. Among all enhancements, we achieved a maximum of 90% increase in object detection accuracy than these traditional models.

The successful integration of a Transformer-based model and dynamic thresholding method for drift detection in video data holds significant implications for the field of machine learning and databases. This work showcases that the Transformer architecture, renowned for its prowess in natural language processing tasks, extends its applicability to the realm of video data analysis [13]. By achieving higher accuracy compared to the conventional GAN-based approach, our research demonstrates that Transformers are not confined solely to pure text-based NLP problems, but possess the versatility to handle reduced-level video and image datasets with equal proficiency.

Furthermore, the exploitation of the Transformer's inherent self-attention mechanism unveils a new paradigm in the detection of model drift. This mechanism allows the model to focus on relevant features, capturing nuanced temporal patterns within the video data. This means that the Transformer's ability to discern subtle shifts in video content makes it a formidable tool for maintaining model accuracy over extended periods. Such adaptability is paramount in applications ranging from surveillance systems to video streaming platforms, where real-time adjustments to evolving data distributions are essential.

In addition, the inherent parallelism harnessed by the Transformer architecture offers a compelling advantage in terms of computational efficiency [17]. This translates to a substantial reduction in both time and monetary costs associated with model training and drift detection. The ability to process video data in parallel not only expedites the training process but also enables the handling of larger and more complex datasets. This, in turn, empowers researchers and practitioners to tackle video analysis tasks at a scale previously considered impractical.

2 RELATED WORK

In the scope of this related work section, our emphasis is specifically directed toward the detection aspect of concept drifting, avoiding the discussion that encompasses both detection and handling strategies. Although various methodologies address the challenge of concept drifting by incorporating mechanisms to detect and adapt to changing patterns, our focus remains centered on methods dedicated to identifying drifting phenomena. Such methods will not be exhaustively explained within this context.

LOF. Introduced by Breunig et al., Local Outlier Factor (LOF) [3] is a density-based method designed to identify outliers by assessing the local density deviation of data points within a dataset. By considering the ratio of a data point's local density to that of its neighbors, LOF excels in capturing anomalies in regions characterized by varying data density. Researchers have leveraged LOF's ability to discern outliers in datasets with heterogeneous density patterns, making it a valuable tool for uncovering anomalies amidst complex and dynamic data structures.

DRAE. Discriminative Reconstruction Autoencoder (DRAE), presented by Xia et al., offers an approach by harnessing the reconstruction errors of an autoencoder, DRAE adeptly discerns inliers from outliers within low-dimensional representations.DRAE enhances discriminative power by infusing self-learned discriminative information into the autoencoder's learning process. This involves a nuanced approach—rather than minimizing errors across all data, emphasis is placed on minimizing errors from positive examples. DRAE iteratively categorizes data as "positive" or "outlier" based on their reconstruction errors, concurrently refining network parameters to yield more discriminative reconstructions.

DA-GAN. Dual-Adversarial GAN (DA-GAN) [22] is a novel method that merges the strengths of an adversarial AE and a GAN. DA-GAN functions as a distance-preserving projection technique, mapping images to a low-dimensional latent space. Its components—encoder, decoder, latent discriminator, and image discriminator—play vital roles in refining the latent space and enhancing image reconstruction. Adversarial discriminators enforce dual constraints, ensuring a smoother latent space without holes and high-quality encoding with minimal information loss.

AUROC Presented by Hendrycks et al.„ Area Under the Receiver Operating Characteristic curve (AUROC) [12] is a threshold independent measure[5], graphically represented by the ROC curve, which depicts the trade-off between true positive rate and false positive rate. Interpretatively, the AUROC signifies the probability that a positive instance exhibits a higher detector score than a negative instance. A random positive detector yields a 50% AUROC, while a perfect classifier attains 100%. Acknowledging potential limitations of AUROC, they also employ the Area Under the Precision-Recall curve (AUPR) as a complementary evaluation metric. Which addresses the base rate issue by plotting precision against recall and provides an informative perspective, especially when class imbalances exist.

MD3. The Margin Density Drift Detection (MD3) methodology [20] is introduced as a novel approach for concept drift detection in streaming data. Unlike traditional methods relying on labeled data, MD3 monitors changes in the margin density of robust classifiers like Support Vector Machines. The margin, representing a classifier's uncertainty, is crucial for generalization over unseen data. MD3 focuses on detecting changes in margin density, considering it an indicative factor of Non-Stationarity. The methodology is application and classifier independent, operating solely on unlabeled data, making it a viable substitute for explicit labeled drift detection techniques.

CBCDD. Critical Blindspot Cardinality Drift Detection,presented by Sethi et al. [21],is designed for detecting concept drift in high

dimensional data streams. The key innovation lies in tracking the average number of samples within critical classification blindspots over time, serving as a robust signal for drift in streaming data environments. Importantly, CBCDD is distribution-independent, classifier-independent, and operates on unlabeled data, making it versatile and applicable in various scenarios.

PCA. Qahtan et al. [19] introduces a novel framework for detecting abrupt changes in unlabeled multidimensional data streams, employing Principal Component Analysis (PCA). Through the projection of data onto selected Principal Components (PCs), univariate streams are formed, effectively capturing variations associated with changes in underlying data variables such as mean, variance, and correlations. The distinctive feature of this approach lies in the independent monitoring of these streams, enabling efficient change detection.

Resampling. Unlike past methods Harel et al. [11] introduces a flexible and robust approach for detecting changes in prediction problems, accommodating various types of concept drift by leveraging random permutations of examples for multiple train-test splits. This method focuses on detecting concept changes within a specified hypothesis class, mitigating false alarms for irrelevant changes. This distinctive approach, previously explored mainly in classification contexts, prioritizes identifying target concept changes through predictor errors.

3 SYSTEM OVERVIEW

3.1 Model Selection

In the realm of computer vision, selecting the appropriate model architecture is paramount for the success of various tasks. Convolutional Neural Networks (CNNs), which have been foundational in image-related tasks, are adept at capturing local patterns and spatial hierarchies within images. However, their sequential processing limits parallelization, hindering their ability to efficiently capture global context. In contrast, Transformers, initially designed for natural language processing, have recently demonstrated prowess in computer vision with models like the Vision Transformer (ViT) and Swin Transformer [16].

Transformers leverage self-attention mechanisms to capture global dependencies effectively, allowing for parallelized processing and a more comprehensive understanding of relationships across the entire input sequence. The Vision Transformer, for example, transforms an image into a sequence of flattened 2D patches, enabling the model to process global contextual information efficiently. This global understanding is particularly advantageous in scenarios where object detection requires capturing relationships between entities that may span the entirety of the input data.

In the context of model drift detection, there is a growing interest in leveraging the capabilities of transformers, driven by findings from recent studies. While the application of transformers in computer vision tasks has traditionally been associated with tasks like image classification and segmentation, their potential for detecting subtle shifts in data distribution has been hinted at in the literature. Previous research, such as the work on DETR (DEtection Transfomer) [4], demonstrated the effectiveness of transformers in achieving state-of-the-art results in object detection tasks. The ability of transformers to consider global context, as evidenced in these studies, raises the hypothesis that they might outperform CNNs in scenarios where detecting changes in the overall scene is critical.

Drawing inspiration from these findings, our exploration into the choice of model architecture for model drift detection is motivated by the notion that transformers' inherent global understanding could provide a more nuanced perspective on evolving patterns in the data. The hope is that this hypothesis, supported by insights from previous transformer-based models in computer vision, will hold true in the specific context of model drift detection. By embracing transformers, we anticipate a more effective detection of subtle and widespread changes in data distribution, ultimately contributing to enhanced performance in scenarios where monitoring and adapting to evolving global patterns are imperative.

3.2 Dynamic Threshold

We introduce a novel approach to efficiently detect model drift - the dynamic threshold method. This method innovatively leverages key metrics, including training probability and information entropy, adapting thresholds in real-time to accurately detect model drift during model prediction and evaluation. Such adaptability is crucial for deep-learning models, particularly in response to the ever-changing data distributions encountered in real-world environments.

The core idea of the dynamic threshold method is rooted in the understanding that a deep learning model while being trained on a dataset, does not merely learn the features of the input but also assimilates essential information about the data's distribution and the model's certainty in its predictions. This insight allows us to harness these learned metrics as benchmarks. We establish a threshold t, which serves as a dynamic reference point. When the metrics derived from real-world data significantly deviate from this threshold—indicative of a change in data distribution or model certainty—we flag this as an instance of model drift.

Therefore, The challenge, and our focus, lies in determining and selecting the appropriate threshold t. This threshold must be dynamic, and capable of adjusting to evolving data patterns while maintaining a balance between sensitivity to drift and stability against false alarms. Our method involves a systematic approach to calibrate and adapt t, ensuring it accurately reflects the model's learning and prediction landscape. By continuously monitoring and adjusting t based on real-time data, our method promises enhanced responsiveness and precision in drift detection, thereby significantly improving the robustness and reliability of deep-learning models in dynamic environments.

Training Probability. An intuition is to use the average maximal trained probability as the threshold to detect model drift. The benchmark X is calculated as follows:

$$\bar{X} = \frac{\sum_{i=1}^{n} max(P(x_i))}{n}$$

where x_i is the 1D probability vector for each prediction made during the last epoch of training. The concept of training probability is closely tied to the notion of model confidence. In probabilistic terms, it can be viewed as the model's estimated probability that a given input belongs to a certain class. This is particularly evident in classification tasks, where the model assigns a probability to each class for a given input, reflecting its certainty or confidence in that

classification. From a theoretical standpoint, training probability is a manifestation of the model's understanding of the data it has been trained on. A low probability may suggest areas where the model is uncertain, potentially due to a lack of representative data or more complex patterns that are harder for the model to learn. Either of these scenarios can indicate instances of model drift.

However, while training probability seems to be a good indicator of model drift, it is not without its limitations. One of the primary drawbacks is its susceptibility to overfitting. During EDA, we discovered that some models (CNN), tend to assign very high probabilities to predictions, even when all inputs in the testing data were unseen drifted data, as shown in Figure 1. This phenomenon was particularly pronounced in scenarios where the model had achieved a high degree of overfitting to the training data. Such overfitting led to an inflated sense of confidence in the model's predictions, as reflected in the training probabilities, which in turn could indicate the emergence of model drift.

Therefore, it is necessary for us to discover a more reliable threshold metric that can improve the accuracy of drift detection.

Information Entropy. A more robust method to detect model drift is to use information entropy as the benchmark metric. In information theory, information entropy measures the uncertainty or randomness in the model's predictions. Therefore, it provides a quantifiable way to assess the predictability and the confidence of a model in its prediction. The metric $E(\bar{x})$ is calculated as follows:

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log_b P(x_i)$$

$$E(\bar{X}) = \frac{\sum_{i=1}^{m} H(X_i)}{m}$$

Where $H(X)$ is the information entropy of a single prediction, and $E(\bar{x})$ is the average entropy of all predictions during the last epoch of training. An entropy value closer to 0 indicates the model's high confidence in its prediction, while an entropy value farther away from 0 implies increased uncertainty.

The integration of information entropy into our drift detection method is based on the assumption that significant changes in entropy can indicate shifts in the model's understanding of the underlying data distribution. A stable model on consistent data typically exhibits relatively consistent entropy values, but as the data begins to drift and drifts are introduced, the entropy is likely to increase, reflecting the model's decreased confidence in its predictions.

The advantage of using information entropy lies in its stability and potential resistance to overfitting. Unlike training probability, which can be overly optimistic in overfitted models, entropy provides a more grounded and less variable measure. As shown in Figure 2, when all inputs in the testing data were unseen drifted data, the information entropy of all is less skewed than training probability shown in Figure 1. Therefore, information entropy serves as a critical component of our dynamic threshold method, offering a stable and reliable metric that enhances our ability to detect and respond to model drift effectively.

3.3 Algorithm

Overview. After completing the model training phase, our algorithm initiates by acquiring training probabilities, represented as P. This is accomplished by utilizing the trained model to predict the training dataset. Subsequently, we apply the information entropy function to each probability p within P. The next step involves computing the mean of all p values, establishing a dynamically determined threshold. We then employ the probabilities obtained from the test set, denoted as $\hat{P}$. For each $\hat{p}$ in $\hat{P}$, we calculate the entropy value $\hat{e}$. If $\hat{e}$ exceeds our threshold, we identify it as an outlier in the result instead of treating it as a classification problem. Otherwise, we consider it an inlier and assign it the predicted label $\hat{l}$.

Algorithm 1 Outlier Detection using Dynamic Thresholding

$P \leftarrow$ **train_model()** {P: probabilities}
$(\hat{P}, L) \leftarrow$ **test_model**(n) {$\hat{P}$: probabilities, L: labels, n: percentage of outliers}
$\hat{L} \leftarrow []$ {$\hat{L}$: predicted labels}
$threshold \leftarrow$ Mean(Entropy(P))
for $\hat{p} \in \hat{P}$ **do**
  $\hat{e} \leftarrow$ Entropy($\hat{p}$)
  if $\hat{e} > threshold$ **then**
    Add **outlier** to $\hat{L}$
  else
    Add $\hat{l}$ to $\hat{L}$
  end if
end for

4 EXPERIMENT

4.1 Experiment Setup

Development. In our endeavor to address model-drift detection, we undertake the development of our very own model-drift detection models, specifically the CNN-based models and Transformer-based models, using Python 3.10.12. To empower these models, we harnessed the capabilities of PyTorch 2.1.0 [18] and TensorFlow 2.14.0 [1], leveraging their extensive features and libraries for efficient model development and training.

Hardware. Our experimentation took place within the Google Colab environment, where we had access to a Nvidia T4 GPU equipped with 12 GB of RAM. Additionally, the underlying hardware infrastructure featured an Intel(R) Xeon(R) processor clocked at 2.2 GHz, also furnished with 12 GB of RAM.

Datasets. The evaluation of our model-drift detection models involved the utilization of two datasets, each chosen to test and validate the effectiveness of our models in different scenarios.

1) MNIST: This dataset comprises a collection of 60,000 28 × 28 black-and-white images, each depicting handwritten digits [6]. For our initial benchmarking, we selected MNIST due to its lower computational resource requirements. Our primary objective with MNIST was to assess the ability of our model-drift detection models to accurately identify drifts as they occur.

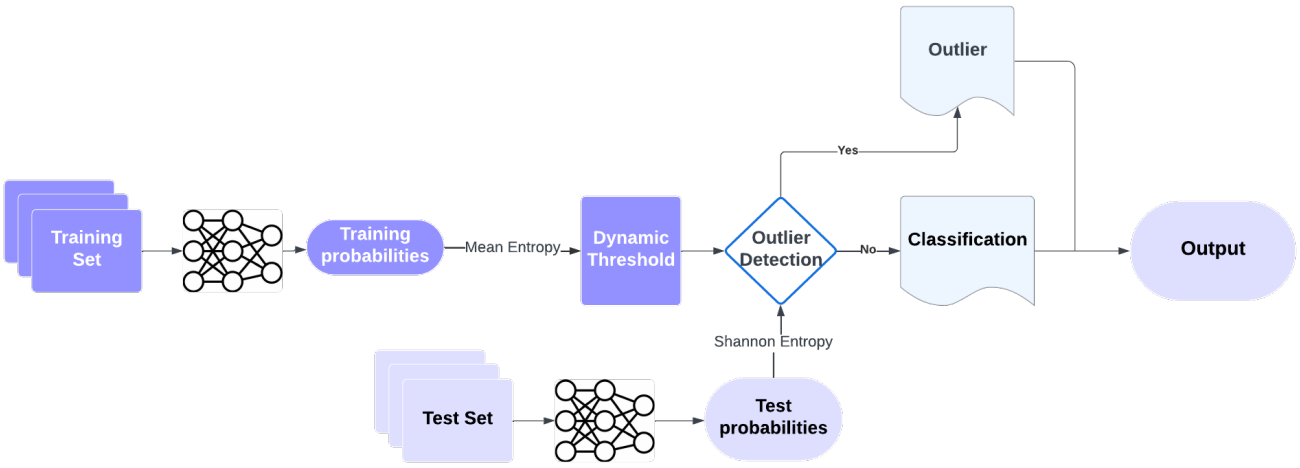

Figure 1: Illustration of Our Proposed Approach Architecture.

2) CIFAR-10: This dataset, consisting of 60,000 32 × 32 colored images distributed across ten distinct classes, served as another crucial dataset for our experiments [14]. CIFAR-10 offered a different challenge and allowed us to test the effectiveness of our model-drift detection models in a diverse visual data setting.

Dimensionality. It's worth noting that while there exist advanced techniques for detecting and handling model drift in low-dimensional data, our approach differs when dealing with high-dimensional data, such as images. The dimensionality of each dataset is determined by the number of pixels in each image. For the MNIST dataset, the dimensionality of images is 784 (28 × 28 pixels) with one channel. Meanwhile, for the CIFAR-10 dataset, the dimensionality of images is 1024 (32 × 32 pixels) with three channels. These varying dimensionality settings provide a comprehensive framework for assessing our model-drift detection models' ability to cope with diverse data complexities, emphasizing the importance of rigorous testing and evaluation in various scenarios.

4.2 Evaluation

Model Selection. In our study, we introduced a novel approach to tackle the challenge of model drift detection. We incorporate VIT[7], the transformer based architecture along with our dynamic thresholding technique, a combination that hasn't been explored before in this context. To evaluate the effectiveness of our method, we plan to compare our detector with well-known baseline detectors like LOF [3]. Additionally, we created a LeNet based traditional Convolutional Neural Network model with our dynamic thresholding technique. This CNN model acts as a starting point for comparison, helping us understand how well the transformer perform in model drift detection compare to the CNN architecture.

Data Modification. In order to simulate real-world scenarios where noisy data or outliers are commonplace, the data used in this experiment was intentionally modified to include instances of both in-scope and out-of-scope data. The datasets chosen for this study were MNIST and CIFAR-10, widely used benchmarks in the field of computer vision.

In the context of the MNIST dataset, our experimental design involves executing the test set across all numerical digits, while selectively training our model drift detector on a subset of these digits. Specifically, we may choose to train the detector on digits 0 through 7, while administering the entire test set, encompassing digits 0 through 9. Consequently, digits 0 through 7 are considered in-scope data, while digits 8 and 9 are deliberately treated as out-of-scope data. Analogously, a parallel procedure is applied to the CIFAR-10 dataset, wherein certain classes are intentionally excluded from the training set based on their labels. This emulates a deliberate departure from the conventional training regimen.

Outlier Adjustment. Aligned with the methodology employed in the ODIN [22] experiment, our investigation entails assessing the performance of the drift detector across various proportions of outlier instances. This systematic exploration involves introducing different percentages of outliers and subsequently scrutinizing the detector's efficacy in identifying and categorizing instances that deviate from the expected data distribution. This approach enables a comprehensive analysis of the detector's resilience and adaptability to diverse outlier scenarios in order to test its robustness under real-world conditions.

4.3 Metrics

In assessing the performance of our drift detector, we will employ two key metrics: Overall Accuracy and Detection Accuracy.

Overall Accuracy. The Overall Accuracy metric takes into account all the correct predictions, this include both in-domain and correctly classified samples as well as out-of-domain samples that are accurately identified as drift instances. The formula for Overall Accuracy is given by:

$$\text{Overall Accuracy} = \frac{\text{\# of correct predictions}}{\text{\# of total samples}}$$

Detection Accuracy. Detection Accuracy specifically focuses on the ability of the detector to correctly identify drift instances among all negative or out-of-domain samples. The formula for Detection Accuracy is:

$$\text{Detection Accuracy} = \frac{\text{\# of detected outliers}}{\text{\# of all outliers}}$$

4.4 Results

In our experimentation, we employ the Vision Transformer (VIT) model alongside LeNet as our transformer and CNN models, respectively. To perform a fair comparison with existing work, we also incorporated LOF (Local Outlier Factor), a common outlier detection technique, as the benchmark to run in our experiment. These models were evaluated on both the MNIST dataset and the CIFAR-10 dataset. In both datasets, the models were trained on the first 8 categories and tested on all categories. We fixed the size of the testing set to 4000 data points and manipulated the percentage of outliers as an independent variable to investigate its influence on both detection accuracy and drift accuracy. For instance, when the percentage of outliers is 20%, we would randomly sample 3600 instances from categories 0-7 and 400 instances from categories 8-9 to form the testing set.

In Table 2, we can derive some noteworthy observations. Firstly, both VIT and LeNet outperform LOF in terms of detection accuracy by a significant margin, indicating their capabilities to accurately capture outliers. Also, we can see that the drift detection accuracy of VIT and LeNet tends to stay consistent across all percentages of outliers, while the detection accuracy of LOF shows a downward trend with the increase of the independent variable. This phenomenon suggests that VIT and LeNet are more robust and stable in outlier detection across varying conditions. In contrast, LOF's performance appears to be more sensitive to the presence of outliers, potentially due to its reliance on local neighborhood information which can be disproportionately affected by outlier data.

Moreover, while the LeNet model achieves some impressive detection accuracies, indicating its proficiency in correctly identifying almost all outliers, its overall accuracy is considerably lower than that of the VIT model. This suggests that the LeNet model tends to detect drift aggressively, frequently mislabeling in-domain samples as out-of-domain, leading to a higher-than-expected false positive rate for outlier detection. Conversely, the VIT model demonstrates a superior overall accuracy, signifying its efficacy in distinguishing between in-domain and out-of-domain samples while performing correct predictions for inliers at the same time. Consequently, the VIT model maintains a commendable 0.95 detection accuracy while achieving an overall accuracy of 0.90.

4.5 Error Analysis

CNN on MNIST. One particular interesting observation we made was that although LeNet or other CNN models perform well on image classification, they generally performed surprisingly poor in respond to outliers with our proposed dynamic threshold. As shown in Figure 2, tends to detect outliers aggressively, resulting in a high false positive value in drift detection.

Unlike VIT, where inliners usually exhibit a fairly uniform distribution across all false classes and a moderate peak in predicted class in softmax scores, LeNet showcases a heightened confidence level for the predicted class in majority of inliners. While this increased probability could be beneficial for classification tasks by ensuring high certainty, it may adversely affect outlier detection. The elevated confidence level raises the probability of establishing a lower threshold from the training set, making it more challenging for unseen samples from the test set to achieve a lower entropy score. Consequently, these samples are more likely to be identified as outliers, even if they shouldn't be.

Outliers	MNIST			CIFAR-10		
	LOF	VIT	LeNet	LOF	VIT	LeNet
0%	0.900	0.898	0.509	0.900	0.312	0.453
10%	0.824	0.892	0.562	0.822	0.334	0.454
20%	0.738	0.897	0.613	0.739	0.359	0.496
30%	0.653	0.913	0.656	0.664	0.372	0.536
40%	0.566	0.917	0.708	0.585	0.398	0.582
50%	0.473	0.910	0.748	0.513	0.423	0.620

Table 1: Overall Accuracy on MNIST and CIFAR-10 using dynamic threshold

Outliers	MNIST			CIFAR-10		
	LOF	VIT	LeNet	LOF	VIT	LeNet
0%	N/A	N/A	N/A	N/A	N/A	N/A
10%	0.118	0.930	1.000	0.110	0.680	0.818
20%	0.094	0.944	0.999	0.098	0.691	0.829
30%	0.088	0.951	0.996	0.107	0.702	0.815
40%	0.083	0.966	0.996	0.106	0.707	0.824
50%	0.069	0.948	0.997	0.113	0.71	0.837

Table 2: Detection Accuracy on MNIST and CIFAR-10 using dynamic threshold

ViT on CIFAR-10. In our exploration of the Vision Transformer (ViT) on the CIFAR-10 dataset, we have encountered notable challenges, despite implementing data augmentations and diverse performance enhancing techniques. The model's suboptimal performance, in comparison to its success on the MNIST dataset, leads us to speculate on a potential deficiency in ViT's inductive bias toward local relationships, particularly in lower layers. This speculation aligns with findings by Zhu et al., which posits that ViT struggles to grasp local relations effectively when faced with small, intricate datasets.[27] Such limitations hinder the model's ability to learn the complexities of CIFAR-10 adequately.

In an effort to address these challenges, we investigated transfer learning by utilizing pre-trained ViT models trained on ImageNet and fine-tuning them for CIFAR-10. While achieving an impressive prediction accuracy of approximately 95%, our evaluation method revealed concerns about the model's detection rate. We attribute this to the pre-trained model's extensive exposure to diverse classes and images during its training on ImageNet. This exposure seems to contribute to the model's reluctance and lack of confidence in predictions, even when accurate. Consequently, this lack of confidence poses a hurdle for our detection method, impeding its effectiveness in identifying instances of model drifting.

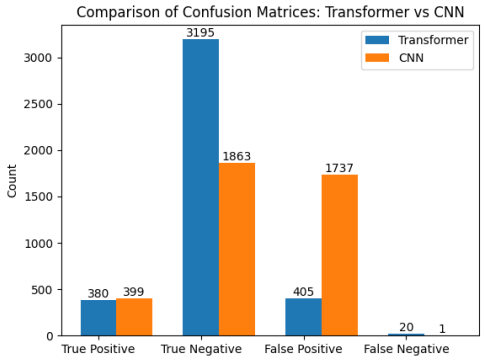

Figure 2: Confusion Matrices Comparison for LeNet and VIT on MNIST

5 DISCUSSION

5.1 Current Limitations

Data Complexity. Our present detector manifests commendable efficacy on tasks characterized by simplicity, as evidenced by its notably superior performance on the MNIST dataset in comparison to CIFAR-10. This performance discrepancy prompts speculation on the dataset's role, suggesting that the detector excels in scenarios involving simpler, grayscale images such as those present in MNIST. The intricate nature of CIFAR-10's color images, with a greater diversity of features and patterns, may introduce complexities that challenge the current detector's adaptability.

Training Cost. An additional constraint surfaces in the form of training cost, which constitutes a substantial limitation. Specifically, the training of our detector, leveraging the Vision Transformer (VIT) model, necessitates a significantly more extensive temporal commitment compared to the training of a conventional Convolutional Neural Network (CNN) model. This increased training time poses practical challenges, potentially impeding the scalability and efficiency of our proposed detection framework in scenarios where computational resources are constrained. Addressing these limitations is integral to refining the detector's applicability across a spectrum of tasks and datasets.

5.2 Future Works

Larger datasets. While our current findings showcase the promising performance of the transformer model, particularly the Vision Transformer (VIT), it is imperative to acknowledge that transformers generally exhibit enhanced capabilities with larger datasets. The modest scale of MNIST and CIFAR-10, though suitable for demonstrating the effectiveness of our detector, limits the full exploration of the transformer's potential. Therefore, a pivotal avenue for future work involves conducting experiments on substantially larger datasets. This endeavor will not only provide a more comprehensive evaluation of the transformer's performance but also facilitate a deeper understanding of its scalability and generalization across diverse and expansive data domains. The outcomes of such investigations will contribute valuable insights into the optimal use of transformer architectures in real-world, large-scale applications.

Leveraging Pretrained Models. An intriguing prospect for advancing our research lies in the integration of pre-trained transformer models. Pre-trained models, having been exposed to extensive and diverse datasets during their training phase, often possess heightened robustness and a broad understanding of complex patterns. Leveraging pre-trained transformer architectures, such as those trained on large-scale datasets like ImageNet, presents an opportunity to imbue our detector with a richer knowledge base. By fine-tuning these pre-trained models on the specific tasks associated with drift detection in MNIST and CIFAR-10, we anticipate a potential enhancement in the detector's adaptability and performance. This approach aligns with the current trend in leveraging pre-trained models to bolster the efficiency and effectiveness of machine learning applications across various domains.

Handling Model Drifts. An effective workflow for managing model drift includes post-detection steps such as automating drift resolution to preserve model accuracy and dependability over time. When a significant drift is identified, the system should initiate an automatic retraining protocol using the most current data, ensuring ongoing relevance and precision. This concept of retraining is not new and aligns with methodologies explored by various researchers, including those from the ODIN team [22]. Additionally, a feedback loop from the end-users can provide valuable insights into how the model is performing in real-world scenarios, allowing for more targeted adjustments and improvements. This comprehensive approach ensures not only the detection but also the effective management of model drifts, thereby enhancing the overall robustness and reliability of the system.

6 CONCLUSION

Detecting drift in data is a significant realm in machine learning both in academia and industry production. Our research presents an innovative approach to model drift detection, combining Vision Transformer (ViT) and LeNet architectures with dynamic thresholding, and demonstrates its efficacy using the MNIST and CIFAR-10 datasets. This novel method not only outperforms traditional techniques like Local Outlier Factor (LOF) but also unveils the potential of transformer models in machine learning challenges. The findings emphasize the importance of adaptable and robust methods in handling dynamically evolving data landscapes in the real-world setting, paving the way for future work to explore more complex datasets and real-time adaptation in diverse domains. This study marks a meaningful advancement in model drift detection, offering a foundation for further research and practical applications in dynamic data environments.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: A System for Large-Scale Machine Learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation* (Savannah, GA, USA) *(OSDI'16)*. USENIX Association, USA, 265–283.

[2] Albert Bifet and Ricard Gavaldà. 2007. Learning from Time-Changing Data with Adaptive Windowing. *Proceedings of the 7th SIAM International Conference on Data Mining* 7. https://doi.org/10.1137/1.9781611972771.42

[3] Markus Breunig, Peer Kröger, Raymond Ng, and Joerg Sander. 2000. LOF: Identifying Density-Based Local Outliers. *ACM Sigmod Record* 29, 93–104. https://doi.org/10.1145/342009.335388

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-End Object Detection with Transformers. *ArXiv* abs/2005.12872 (2020). https://api.semanticscholar.org/CorpusID:218889832

[5] Jesse Davis and Mark Goadrich. 2006. The Relationship between Precision-Recall and ROC Curves. In *Proceedings of the 23rd International Conference on Machine Learning* (Pittsburgh, Pennsylvania, USA) *(ICML '06)*. Association for Computing Machinery, New York, NY, USA, 233–240. https://doi.org/10.1145/1143844.1143874

[6] Li Deng. 2012. The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Processing Magazine* 29, 6 (2012), 141–142. https://doi.org/10.1109/MSP.2012.2211477

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *CoRR* abs/2010.11929 (2020). arXiv:2010.11929 https://arxiv.org/abs/2010.11929

[8] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Hamid Bouchachia. 2014. A Survey on Concept Drift Adaptation. *ACM Computing Surveys (CSUR)* 46 (04 2014). https://doi.org/10.1145/2523813

[9] João Gama, Ricardo Fernandes, and Ricardo Rocha. 2006. Decision Trees for Mining Data Streams. *Intell. Data Anal.* 10, 1 (jan 2006), 23–45.

[10] Vinicius Goncalves, Lourival Silva, Fátima Nunes, João Ferreira, and Luciano Araujo. 2023. Concept drift adaptation in video surveillance: a systematic review. *Multimedia Tools and Applications* (06 2023), 1–41. https://doi.org/10.1007/s11042-023-15855-3

[11] Maayan Harel, Koby Crammer, Ran El-Yaniv, and Shie Mannor. 2014. Concept Drift Detection through Resampling. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32* (Beijing, China) *(ICML'14)*. JMLR.org, II–1009–II–1017.

[12] Dan Hendrycks and Kevin Gimpel. 2018. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. arXiv:1610.02136 [cs.NE]

[13] Daniel Kang, Peter Bailis, and Matei Zaharia. 2019. BlazeIt: Optimizing Declarative Aggregation and Limit Queries for Neural Network-Based Video Analytics. arXiv:1805.01046 [cs.DB]

[14] Alex Krizhevsky. 2009. Learning Multiple Layers of Features from Tiny Images. https://api.semanticscholar.org/CorpusID:18268744

[15] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. 2022. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision*. Springer, 280–296.

[16] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021), 9992–10002. https://api.semanticscholar.org/CorpusID:232352874

[17] Julian Richard Medina and Jugal Kalita. 2018. Parallel Attention Mechanisms in Neural Machine Translation. *University of Colorado Colorado Springs Journals* (2018). https://arxiv.org/pdf/1810.12427.pdf

[18] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf

[19] Abdulhakim A. Qahtan, Basma Alharbi, Suojin Wang, and Xiangliang Zhang. 2015. A PCA-Based Change Detection Framework for Multidimensional Data Streams: Change Detection in Multidimensional Data Streams. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Sydney, NSW, Australia) *(KDD '15)*. Association for Computing Machinery, New York, NY, USA, 935–944. https://doi.org/10.1145/2783258.2783359

[20] Tegjyot Singh Sethi and Mehmed Kantardzic. 2017. On the Reliable Detection of Concept Drift from Streaming Unlabeled Data. arXiv:1704.00023 [stat.ML]

[21] Tegjyot Singh Sethi, Mehmed Kantardzic, and Elaheh Arabmakki. 2016. Monitoring Classification Blindspots to Detect Drifts from Unlabeled Data. In *2016 IEEE 17th International Conference on Information Reuse and Integration (IRI)* (Pittsburgh, PA, USA). IEEE Press, 142–151. https://doi.org/10.1109/IRI.2016.26

[22] Abhijit Suprem, Joy Arulraj, Calton Pu, and Joao Ferreira. 2020. ODIN: Automated Drift Detection and Recovery in Video Analytics. arXiv:2009.05440 [cs.CV]

[23] Hans Thisanke, Chamli Deshan, Kavindu Chamith, Sachith Seneviratne, Rajith Vidanaarachchi, and Damayanthi Herath. 2023. Semantic segmentation using Vision Transformers: A survey. *Engineering Applications of Artificial Intelligence* 126 (2023), 106669.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[25] Yan Xia, Xudong Cao, Fang Wen, Gang Hua, and Jian Sun. 2015. Learning Discriminative Reconstructions for Unsupervised Outlier Removal. *2015 IEEE International Conference on Computer Vision (ICCV)* (2015), 1511–1519. https://api.semanticscholar.org/CorpusID:13982294

[26] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. 2021. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision*. 558–567.

[27] Haoran Zhu, Boyuan Chen, and Carter Yang. 2023. Understanding Why ViT Trains Badly on Small Datasets: An Intuitive Perspective. arXiv:2302.03751 [cs.CV]