The New-Normal of Fake News: Enhanced and Sustainable Multi-Expert Detection through Timely Adaptation to Counter Knowledge Obsolescence

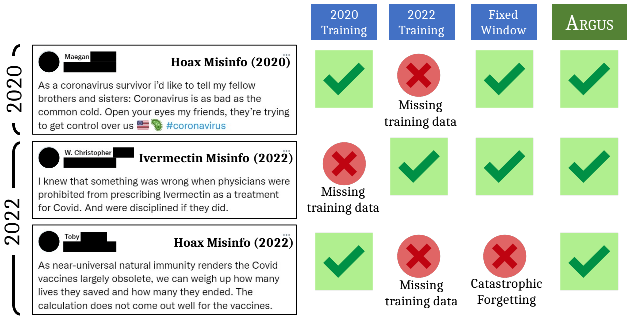
Anonymous submission

Abstract

The explosive growth of misinformation in online communities reflects the accelerating introduction of unprecedented changes in the real world, e.g., the new misinformation campaigns in the evolving COVID-19 pandemic. We call such data streams on unprecedented changes the *new-normal*, which have three distinguishing properties: novelty, prevalence, and ephemerality. Using a large-scale COVID fake news data set spanning 25 months, as well as 11 public fake news datasets across multiple modalities, we show the existence and importance of new-normal. The three properties of the new-normal reduce the performance of static classifiers trained on fixed ground truth, a phenomenon we call knowledge obsolescence (KO), confirmed in extensive experiments. A demonstration adaptive system, Argus, shows the feasibility of recovering from KO through continuous adaptation by generating new submodels and adjusting their weights according to KO measurements. Over 25 months of evolving COVID fake news, Argus's KO detection, dynamic submodel generation, and adaptive submodel selection achieves 2x higher accuracy compared to static classifiers, as well as 1.3x higher accuracy compared to fixed-window expert systems.

1 Introduction

Modern social media have facilitated the spread of misinformation on evolving topics, such as the COVID-19 infodemic (Enders et al. 2020). With impact on mission-critical and life-threatening topics such as vaccine hesitancy, research on accurate and timely detection of misinformation and fake news is urgently needed. As examples of accurate detection, state-of-the-art approaches based on expert systems built around pre-trained language models (PLMs) [ref] with refinements (e.g., social context, and keyword attention networks) have been trained on curated, fixed datasets and tested on data matching the training distribution. Unfortunately, as the infodemic evolves with the pandemic, these state-of-the-art approaches have increasing difficulties as time passes (Suprem and Pu 2022a; Dun et al. 2021), shown in Fig. 1. In retrospect, the decreasing performance of machine learning (ML) classifiers trained from fixed ground truth is not surprising, since new misinformation campaigns are completely unknown to those ML classifiers.



As a coronavirus survivor i'd like to tell my fellow brothers and sisters: Coronavirus is as bad as the common cold. Open your eyes my friends, they're trying to get control over us 🇺🇸 🦠 #coronavirus

I knew that something was wrong when physicians were prohibited from prescribing Ivermectin as a treatment for Covid. And were disciplined if they did.

As near-universal natural immunity renders the Covid vaccines largely obsolete, we can weigh up how many lives they saved and how many they ended. The calculation does not come out well for the vaccines.

Figure 1: Adapting to New-normal: Static classifiers relying on fixed training data can make mistakes on out-of-distribution samples. Fixed window updates models can replace existing knowledge due to catastrophic forgetting. With Argus, we adapt by selecting the best fit classifier for each data point

.

New-Normal. Real-world misinformation such as COVID fake news exhibit three properties that make them challenging for gold standard ML practice of training classifiers from fixed, curated ground truth: novelty, prevalence, and ephemerality. First, the new-normal fake news are novel (and wildly imaginative) when introduced, e.g., COVID caused by 5G signals or cured by ivermectin (Hussna et al. 2021). Next, the new-normal are created and injected into social media en masse (prevalence) during sustained misinformation campaigns. Finally, as their novelty wears off and credibility declines, the misinformation campaign fades out (ephemerality), and is replaced by new campaigns, starting another cycle of the new-normal.

They are at odds with an implicit, but classic, assumption in the gold standard ML evaluation method based on fixed ground truth: that the knowledge is timeless. Typical annotations of ground truth do not include an expiration date. In contrast, the new-normal data sets that exhibit novelty, prevalence, and ephemerality would render classifiers trained on fixed ground truth obsolete over time. We will present experimental evidence showing that the time-dependent variability of knowledge in new-normal fake news data sets will lead to KO in several representative classifiers with state-of-the-art architectres trained with fixed ground truth.

Contribution 1. The first contribution of the paper, presented in Sec. 3, is a methodical demonstration of the reality of new-normal data sets with three properties: novelty, prevalence, and ephemerality. The three properties lead to KO in state-of-the-art, but static classifiers, when evaluated with new test data that was created in a different time than when fixed ground truth was been annotated. We conduct extensive experiments across multiple public datasets, including a large-scale dataset spanning 25 months, to validate existence of new-normal challenges.

Contribution 2. To meet the KO challenges posed by new-normal, we describe a methodical KO recovery process and experimental demonstration (4.1). KO recovery consists of continuous monitoring of obsolesence in a data stream, followed by adaptation actions when significant changes arrive, submodel generation or updates and dynamic submodel selection. We implement these in a classifier management platform we call Argus [1], and use Argus to evaluate KO detection and recovery. On a large-scale fake news detection dataset spanning over 25 months where we evaluate over monthly time windows, Argus achieves overall 2x higher accuracy than static classifiers, and 1.3x higher accuracy than fixed window classifiers.

2 Motivation and Related Work

2.1 Motivation: Reality of New-Normal

The three properties of new-normal have a growing presence in a world evolving at an increasing pace. A classic example that has been explored is Google Flu Trends (GTF), a statistical model of influenza spread using Google search terms data from the 2007-2008 flu season (Dugas et al. 2013). GFT showed 97% initial accuracy, but the model prediction error grew steadily to 100% by 2014 (Lazer et al. 2014), leading to the deactivation of GFT in 2015.

We show new-normal properties in social media misinformation campaigns with the FNC dataset (Suprem and Calton 2022) and COVID-FN (Chen, Lerman, and Ferrara 2020) datasets (described in Sec. 3.1), which comprise misinformation and COVID-related social media posts, respectively, over 25 months during the pandemic. In Fig. 2, we show the quantity of fake news associated with 3 keywords obtained from recent fake-news classifier works: 5g, exposure related misinformation, and ivermectin (Hussna et al. 2021). We show the quantity of fake news for a term relative to its occurence across all 25 months; each keyword's posts are identified by a distinct classifier combining keyword detection, sentiment analysis, and weak supervision adapted from (Li et al. 2021). The novel ivermectin-related misinformation were introduced in 2021, becoming prevalent in Fall 2021, with a resurgence in early 2022, and fading afterwards (ephemerality). Similarly, 5g-related misinformation have multiple peaks as they ebb and flow, and exposure-related misinformation peak twice: during the Delta wave (the first variant of concern) in Fall 2021, aned the Omicron wave in early 2022.

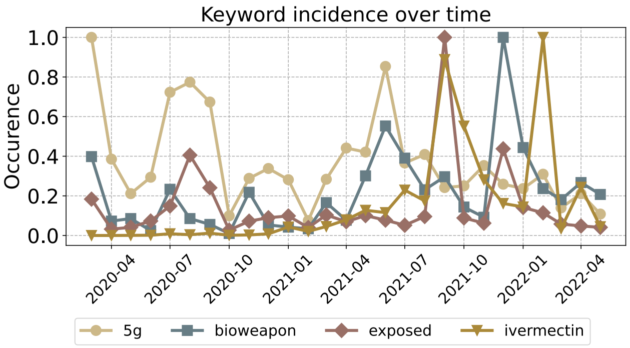


Figure 2: Incidences of different types of misinformation; occurence is calculated separately for each misinformation relative to its total occurence through the pandemic

.

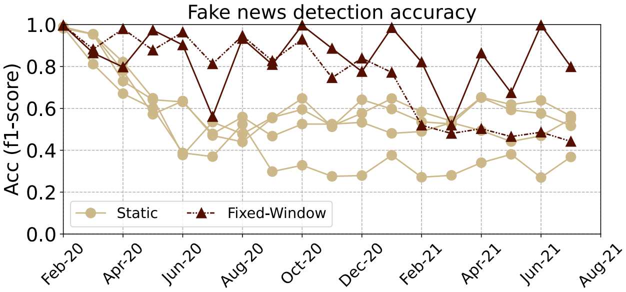


Figure 3: Accuracy declines due to KO. Static classifiers face rapid decline in accuracy. Fixed window update classifiers have more varying accuracy, which is also impractical. Further, consecutive updates can cause catastrophic forgetting (dashed red)

.

Knowledge obsolescence experiment. We provide further quantitative evidence of the impact of new-normal phenomenon with a motivating experiment (described in detail in Sec. 3.1) on FNC, shown in Fig. 3:

1. **Static classifiers:** We train classifiers on first three months of the FNC data. Then we evaluate them on the test set of the entire dataset.
2. **Fixed-window classifiers:** We update classifiers from ❶ on 1-month delayed monthly training data from FNC (the delay simulates labeling time).

It is expected that static classifiers (yellow) would not perform well as new misinformation enters the data stream; However, fixed-window classifiers also have decline in accuracy. With 1-month updates, classifiers face catastrophic forgetting or mode collapse due to too many updates on drifted streams. On the other hand, extending the update window to 3-6 months provides significant gaps in accuracy recovery, leading to higher variance in accuracy. This accuracy deterioration is caused by knowledge obsolescence (KO), which occurs when the prediction data no longer matches the training distribution of an expert. We observe obsolescence early

[1] Argus is named after the Greek mythical figure who is always alert for changes using its many eyes

with the static classifiers, as they fail to adapt to changing misinformation topics. The fixed window classifiers also fail to adapt since they rely on a monotonically changing assumption of fake news topics. However, as we showed in Fig. 2, misinformation reappears throughout the stream in different contexts, so updated classifiers either experience catastrophic forgetting or mode collapse, leading to accuracy deterioration. In this case, the novelty of fake news is not periodic, so fixed-window classifiers miss the fake news campaigns. Even the models trained with the highest quality, but fixed, training data would suffer KO when the environment changes significantly, e.g., when the new-normal show little overlap with the past training data due to ephemerality.

2.2 Related Work

Relative Novelty. Concept drift (Gama et al. 2014) is an area focused on the detection of changes of statistical properties in a data stream. Significant advances have been made in change detection (Suprem et al. 2020) under specific conditions of virtual drift, which assumes limited changes within fixed data sets, or in numeric sensor data (Žliobaitė, Pechenizkiy, and Gama 2016). In contrast, the new-normal are more closely related to real concept drift, where absolute novelty (non-stationary new data) arrive constantly and unboundedly.

Rarity as Novelty (Outliers and Anomalies). Novelty detection algorithms (Pidhorskyi, Almohsen, and Doretto 2018) are excellent in finding rare anomalies due to sparsity in feature space. Because of novelty and prevalence, classic outlier and anomaly detection algorithms would not apply to the new normal data directly.

Ephemerality of New-Normal. If timeless knowledge assumption were applicable, retraining through active learning (Su et al. 2020) or other selective training approaches would allow for quick updates of deployed classifiers. However, the three properties of new-normal lead to multiple, unrelenting waves, overlapping and superseding each other. In Fig. 3, we showed that fixed-window updates using provided training data causes eventually causes catastrophic forgetting. To the best of our knowledge, there are few papers (and data sets) that discuss or study the ephemerality of new-normal data sets and KO.

COVID Fake News. Recent research on detecting COVID misinformation has relied on small-scale datasets collected over a short span of time (Suprem and Pu 2022a). In addition, there are several state-of-the-art approaches for fake news detection given a fixed dataset; we used these approaches for the motivating experiments to show accuracy decline. More recently, KAN (Dun et al. 2021) uses adaptive knowledgebases to improve fake news classification accuracy; by switching the classifier knowledgebase between politics and COVID, KAN improves on fake news detection on different datasets. Our work extends the underlying idea by continuous obsolesence detection, where we monitor a data stream for relevance to the training data (or knowledgebase) and with submodel selection, adaptively select the best-fit submodel for a prediction sample.

3 Detecting Knowledge Obsolesence

3.1 Evaluation Approach

The novelty in new-normal is a major reason for knowledge obsolescence (KO) to cause performance declines in classifiers that achieved excellent performance trained from fixed ground truth. Specifically, the novelty in new-normal changes the joint distribution of test data and true labels $P(X, Y)$. A model trained on an initial distribution X to predict labels Y will be expected to suffer performance degradation when tested on an evolved X', where the intersection $X' \cap X$ is shrinking. This intuitive explanation suggests that the performance drop should happen for a variety of static models whenever the test data set has decreasing overlap with the original training data.

Evaluation Classifiers. To evaluate the scope of KO, we compared three representative families of fake news detectors: plug-and-play BERT, social-context architectures, and multi-input architectures:

- **BERT variants.** Here we have used off-the-shelf, pretrained BERT-based encoders and fine-tuned the classifier heads on different datasets. We used BERT, AlBERT, and Covid-Twitter-BERT.
- **SocialContext.** These architectures combine misinformation topic detection and social context such as likes, retweets, etc, for improved detection accuracy. We used the Fakeddit (Nakamura, Levy, and Wang 2019) approach of combining the raw text with 'like' and 'share' counts, and the NELA (Gruppi, Horne, and Adalı 2022) approach of using username whitelists.
- **MultiInput.** We used MDAWS (Li et al. 2021), where a trained model is combined with weak supervision signals, such as username whitelists, sentiment analysis, and keyword detection (such as swear words).

We perform 2 sets of evaluations on these classifiers to validate existence of KO: cross-dataset evaluation, and single-dataset windowed evaluation

Cross-Dataset Evaluation Datasets. Given n related, e.g, fake news, datasets, we train classifiers on 1 fake news dataset and test on remaining $n-1$ datasets. It is well known that cross-dataset testing leads to lower accuracy due to domain shift. We use $n = 11$ existing fake news datasets, comprising news articles, social media posts, and titles, described in Table 1. All datasets here come from (Suprem and Pu 2022a).

Cross-Dataset, Initial Baseline. We perform some preliminary experiments using evaluation classifiers in each dataset: (i) same-dataset accuracy shows the test accuracy of a classifier trained on the same dataset; (ii) cross-dataset accuracy shows test accuracy of a classifier on remaining non-training datasets, and (iii) similar-dataset accuracy shows accuracy of classifiers when tested on datasets with similar content. For example i.e. we test the ‘k_title‘ classifier on ‘c19_title‘ and ‘miscov‘ since they are both datasets of fake news titles.

We have shown averaged results across classifiers, since performance was similar in each case. Due to decreased overlap between training and testing data, classifiers have lower accuracy compared to same-dataset testing.

Table 1: Training datasets and initial baseline results. Datasets from (Suprem and Pu 2022a). Same-Dataset accuracy tests classifier on corresponding test set. Cross-Dataset tests classifier on remaining datasets' test sets. Similar-Dataset tests classifier on remaining datasets with the same dataset content.

Dataset		**Accuracy**		
Name	**Content**	**Same-Dataset**	**Cross-Dataset**	**Similar-Dataset**
k_title	Titles	0.97	0.60	0.50
coaid	Articles	0.97	0.63	0.60
c19_text	Articles	0.98	0.61	0.73
cq	Tweets	0.54	0.51	0.53
miscov	Titles	0.55	0.50	0.49
k_text	Articles	0.98	0.57	0.56
rumor	Tweets	0.83	0.55	0.52
cov_fn	Tweets	0.96	0.51	0.46
fakeddit	Tweets	0.80	0.64	0.49
nela	Tweets	0.72	0.60	0.53
c19_title	Titles	0.95	0.63	0.49

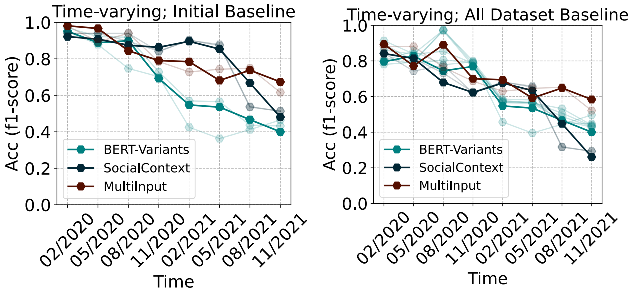


Figure 4: Initial experiments on the FNC dataset with time-varying evaluations.

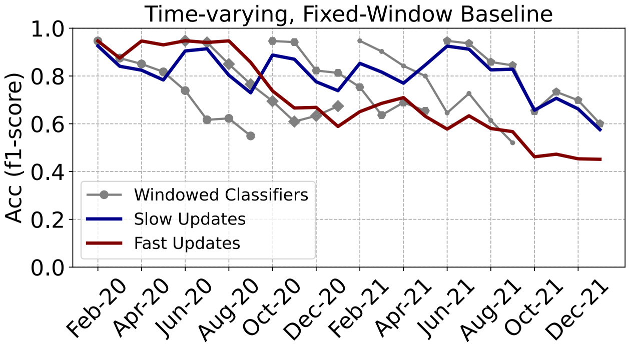


Figure 5: Time-varying, fixed window baseline. Gray lines represent fixed-window models trained on a specific month's training data; we show every 4 months for easy viewing. Slow updates (blue) occur every 4 months. Fast updates (red) occur each month.

.

Time-Varying Evaluation Datasets. Here, we test classifiers in a streaming time-varying setting, where classifiers are initially trained on some preliminary data, then tested across multiple time windows (similar to motivating experiment). We use FakeNewsCovid (FNC), a large-scale, multi-year fake news dataset from (Suprem and Calton 2022) that contains over 1B social media posts consisting of tweets and links to social media posts, videos, and images with captions spanning the COVID-19 pandemic, from January 2020 through July 2022.

Time-Varying, Initial Baseline. As a starting point, we trained BERT, SocialContext, and MultiInput classifiers on the first 3 months of FNC data. We then tested them on each windows of FNC; we show representative experiments in Fig. 4, with remaining experiments as faded lines; we also tested variations trained on the first 3 months plus the training sets of datasets in Table 1, shown in the second graph. These performed slightly worse due to distributional differences in the training sets. The BERT variants show significant performance degradation, leading towards random guess accuracy for most models. SocialContext and MultiInput approaches have slower performance degradation due to additional knowledge used in misinformation prediction. Over time, they also approach random guess accuracy.

Time-Varying, Windowed-Baseline. A common approach for time-varying streaming setting is to update models with new data. Here, we use FNC's monthly training data to perform cumulative updates to the initial baseline classifiers, shown in Fig. 5. We simulate training data lag due to labeling time by using time-delayed training data by 1 month. Monthly updates on models led to catastrophic forgetting where models approached random-guess after a few updates (red in Fig. 5). Slower updates (we show every 4 months in blue) lead to less accuracy decline, but increase variance in accuracy, which is impractical for deployments.

Obsolescence and Accuracy. We can understand the drop in accuracy by observing the relationship between training and test data in each experiment. Intuitively, if test data differs sufficiently from training data, then accuracy can decrease (Suprem and Pu 2022b). We can measure this difference with cluster overlap of the training and prediction features using point proximity on a distance metric; with text features, we can use cosine similarity as a distance metric. That is, given 2 set of samples: training data A and prediction samples B, we calculate, for each sample $x \in B$:

$$P_B(x) = d_{x,B}/d_{x,A} \quad (1)$$

Here, $d_{x,B}$ is the distance between x and the closest point in B, and $d_{x,A}$ is the distance between x and the closest point in A. So, $P_B < 1$ when x is closer to points in the prediction data than to training data. To find the total overlap of B in A, we compute: $|P_B > 1|/|B|$. This is similar to the TrustScore approach (Jiang et al. 2018) that uses class distributions. To improve local representation, we instead use multiple proxies for each class, using KMeans.

Implementation. For each experiment earlier, we compute the overlap between prediction clusters and training clusters of the experiment's classifier, and take the highest overlap

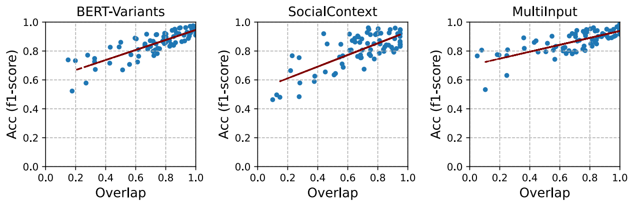


Figure 6: Comparison of overlap computed from point-proximity to accuracy between training and test clusters

value. We compare this overlap value to the accuracy for that prediction cluster, shown in Fig. 6: as overlap decreases, accuracy also decreases across all classifier variants. This is a useful and easy metric for detecting relevance, and thus obsolesence. However, there are drawbacks to directly using the data stream: if we wish to use point-proximity overlap to detect obsolescence, we need to empirically determine an overlap threshold to trigger obsolescence detection. Further, we need sufficient prediction samples in a cluster to compute obsolescence. With these limitations in mind, we propose ModelDrift: an approach to estimate the overlap between prediction and training samples using only a batch of prediction samples.

3.2 ModelDrift: Detecting Obsolescence

We will now present our ModelDrift metric to detect knowledge obsolescence. To compute obsolescence, we need to measure whether prediction sample has diverged sufficiently from training data of a model. We can use research on adversarial attacks as a starting point: adversarial attacks succeed because they exploit this model overconfidence. By slightly perturbing an input sample, adversarial attacks can force models to confidently provide incorrect predictions (Poursaeed et al. 2018). This occurs due to lack of smoothness in the embedding space, i.e. Lipschitzness (Urner and Ben-David 2013). As (Chen et al. 2022) show, models are more accurate when the embedding space around a sample is smooth. Conversely, when the embedding space is not smooth, slight perturbations in the input can significantly deviate the embedding, changing a model's predictions. So, we can use a model's embedding smoothness to determine model overlap with prediction data more accurately as follows.

Lipschitz Smoothness. First, we need to define Lipschitz smoothness. From (Urner and Ben-David 2013), a classifier f is L-Lipschitz smooth if for some label C, distance function d, and any 2 samples with features x_1,x_2:

$$|\Pr(f(x_1) = C) - \Pr(f(x_2) = C)| \leq L \cdot d(x_1, x_2) \quad (2)$$

For any two samples, the difference in prediction probabilities between x_1 and x_2 is bounded by an L factor of the distance between the two samples' features. Since this value of L applies everywhere, actual values of L are large. We can use class definition of L by using L for each class in a model's training data.

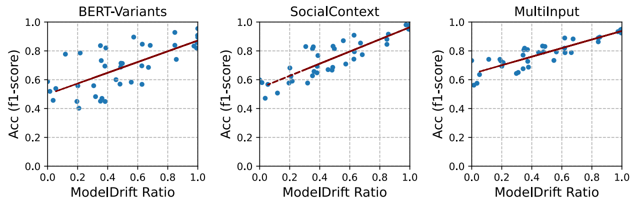


Figure 7: Comparison of ModelDrift Lipschitz ratio to accuracy between training clusters and prediction samples.

We can further localize L by first splitting each class in several proxy clusters with KMeans. Then, for each cluster, we take the cluster centroid as x_1. Using the points inside the cluster, we can estimate L for that cluster using Eq. 2.

L Threshold. Interestingly, the L value for each cluster also gives us a natural threshold for obsolesence detection. Per observations in (Urner and Ben-David 2013), a function that is L-Lipschitz smooth also satisfies the following condition.

Let $\Phi : \Re \rightarrow [0, 1]$. Given x', $x_r \sim P_X$, we say that f is probabilistically Lipschitz, or Φ-Lipschitz if, for all $\epsilon > 0$, there is an increasing function $\Phi(\epsilon)$ such that:

$$\Pr_{x', x_r \sim P_X}[d(f(x'), f(x_r)) - \frac{1}{\epsilon} d(x', x_r) > 0] \leq \phi(\epsilon) \quad (3)$$

That is, $\Phi(\epsilon)$ bounds the probability of f's predicted label changing within a radius ϵ of x'. Tying the notation back to Eq. 2, we note that $\epsilon = 1/L$. Conversely, outside this radius ϵ, the probability of the predicted label increases without bound. So, we can use the ϵ-ball region around x' as an indication of model smoothness.

The approach, then, is as follows. Given x', perturb it uniformly in an ϵ radius, where we compute ϵ using the L value of the nearest training data proxy cluster to x'. Then, compute the $L_{x'}$ value for x' using the perturbations, and compare it to the L value of the nearest training data proxy cluster We can keep model predictions if $L_{x'} \leq L$, and abstain otherwise. This means we do not need to empirically measure a threshold: the computed L value is the threshold for each proxy cluster. Once we have an L value for each cluster, we can detect obsolescence for prediction data if the region around a prediction sample is less smooth than the nearest training data proxy cluster.

Implementation. For each experiment in Sec. 3.1, we use training data proxy clusters and compute L values for each cluster. We take the maximum L value as the upper bound and use it as the threshold, since L is the maximum of all potential values in Eq. 2. Then, for each prediction sample, we compute the prediction embedding smoothness and calculate the ratio between L and $L_{x'}$. We observe accuracy for $L/L_{x'} < 1$; that is, when the perturbed samples are less smooth than the nearest proxy, in Fig. 7.

As the ratio decreases below 1, classifier accuracy decreases, similar to Fig. 6. Here, however, we have an automatic threshold in the ratio of $L/L_{x'}$: whenever the ratio is less than 1, we can trigger obsolescence. This is different from the point-proximity method, since point-proximity is bounded in $[0, 1]$ and we need to find some value in that range to trigger obsolesence, whereas the ratio $L/L_{x'}$ is in $[0, \infty)$.

4 Recovery from KO

In Sec. 3.1, we showed the impact of KO on several representative ML classifiers trained on fixed ground truth, and tested on evolving new-normal fake news data sets. Since the root cause of KO is the never-seen-before novelty of new-normal, a natural question that arises is whether it is possible for any ML classifiers to recover from KO. Fortunately, the answer is positive. We will demonstrate KO recovery with a combination of ModelDrift for obsolesence detection and an adaptive decision module for best-fit submodel selection.

4.1 Argus Demonstration System

We have built a demonstration system, called Argus, to demonstrate recovery from KO. Argus is designed as a modular and adaptive classifier manager using team-of-experts (Pu et al. 2020), where the submodels are created from the evolving new-normal data sets.

Argus comprises of 3 components: (1) the ModelDrift obsolescence detector, (2) new submodel generator, and (3) an adaptive decision module (ADM) for best-fit submodel selection. ModelDrift identifies novelty that would cause submodels to fail due to obsolescence. The submodel generator adds new submodels trained on the novel training data to prepare for prevalence. Finally, the ADM selects the best-fit submodel for prediction samples: by computing overlap between prediction to training data clusters, ADM combined with ModelDrift can quickly determine if Argus needs to use an existing submodel that has high overlap with the prediction sample or trigger new submodel generation.

The limitations of the demonstration system and future work are discussed in Sec. 4.4.

ModelDrift KO Detector. Given a prediction sample, we first obtain features from each submodel. Then, perturb these features in an ϵ radius, where $\epsilon = 1/L$, and L is the Lipschitz threshold computed for each classifier using the nearest training data proxy cluster with a KDTree. Each submodel provides logits for the original features as well as perturbed features. We can compute the perturbation $L_{x'}$ value between logits and features using Eq. 2, and compare this to the submodel's L value. If the ratio is less than 1, the submodel abstains from prediction. If all submodels abstain, we trigger obsolescence detection, since all submodels have mismatch between their training data and the prediction sample.

New Submodel Generation. Once KO is detected, we need to prepare for prevalence of the novel data. We tested two versions of new submodel generation: AlwaysGenerate and ThresholdGenerate. In AlwaysGenerate, we create a new submodel each time we detect obsolescence. In ThresholdGenerate, we check if the $L/L_{x'}$ ratio for some models is above some threshold α. For these models, we will update them on the novel data, since there is some overlap. We compare both approaches in Sec. 4.3.

ADM. If there is at least one submodel with $L/L_{x'} > 1$, ADM selects that submodel for prediction. In the case there are no available submodels due to KO, we still need to provide predictions while new training data is collected and new submodels are being trained. In this case, the ADM selects the top k models with smallest L to provide predictions.

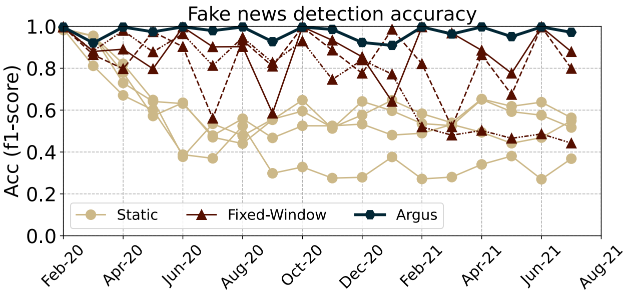


Figure 8: Recovery from KO with ModelDrift and ADM: Argus uses ModelDrift to detect KO, and ADM to select the best-fit submodel to deliver predictions

4.2 Evaluation Results

We evaluate Argus on the time-varying windowed evaluation described in Sec. 3.1. Argus is compared to static classifiers that are trained at the beginning of the FNC stream as well as on the 11 fake news datasets in Table 1. We also compare Argus to the fixed-window approach, where we generate and update new models on monthly, bimonthly, and trimonthly windows and use the most recent update models for each sample. We show results on the FNC dataset in Fig. 8, where we have selected representative experiments from Static and Windowed approach, plus Argus.

The ability of adaptive submodel selection from the team-of-experts implemented in Argus to recover from KO compares favorably to the accuracy decline in static and fixed-window classifiers. Over each experiment, the static approach (yellow plots), regardless of variant, experienced accuracy decline over the course of FNC. While there were some variant-hyperparameter-initialization combination that occasionally had increased performance, we have omitted them here for cleaner visualization. Similarly, the fixed-window approaches (red plots) perform well after updates, but decline in accuracy afterwards. We have shown monthly, bimonthly, and trimonthly up-date windows. In each case, while there is higher average accuracy relative to static classifiers, there is significant variance, reducing their effectiveness. Further, monthly updates lead to catastrophic forgetting in several experiments (we have shown one), reducing a classifier's effectiveness completely.

Argus, on the other hand, maintains high accuracy throughout with small variance, with 2x accuracy versus static classifier's average accuracy, and 1.3x the average accuracy of fixed-window classifiers. By dynamically detecting KO, generating submodels, and implementing an ADM for best-fit model selection, we mitigate training (catastrophic forgetting) and variance issues of a fixed window approach and accuracy degradation of a static approach.

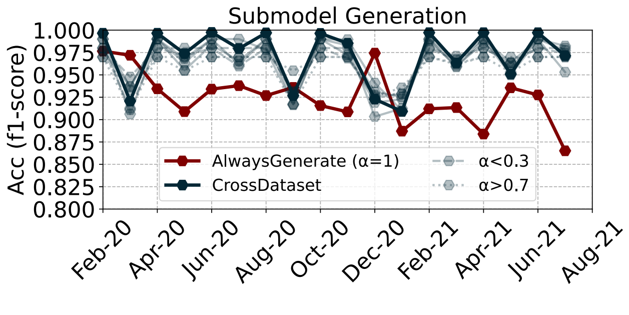


Figure 9: New submodel generation comparing AlwaysGenerate and ThresholdGenerate

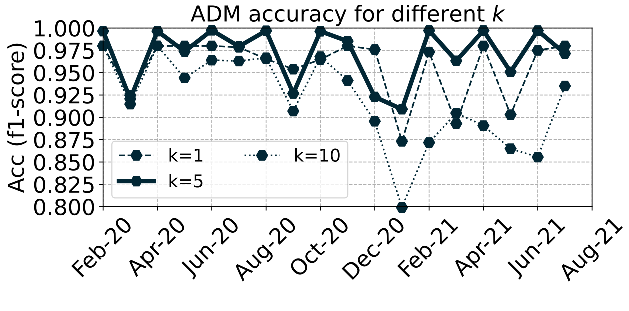


Figure 10: We compare different k values for ADM's top-k submodel selection. With large k, we are including more obsolete submodels, and with small k, we use few sources, yielding noiser predictions.

4.3 Design Choices

New Submodel Generation. . We compare AlwaysGenerate and ThresholdGenerate for Argus' submodel generation policy. For ThresholdGenerate, we use sweep α in $[0, 1]$ as well as a threshold computed from cross-dataset evaluation. We compute this latter threshold as follows: for each classifier, we measure the average L ratio where L-ratio is less than 1 and the classifier correctly classifies a sample from cross-dataset test clusters. In effect, we rely on a submodel's generalizability on out-of-distribution samples to estimate a good L threshold for updating versus generating a new submodel. We compare these approaches in Fig. 9, where the accuracy differences are negligible in ThresholdGenerate unless α is close to the extremes of $[0, 1]$. This is because when α is small, we update submodels more often instead of generating new ones. This leads to catastrophic forgetting. When $\alpha \rightarrow 1$, we update submodels rarely, in effect 'freezing' them. So, submodels do not incorporate new knowledge that is more similar to their training data and do not get fine-tuned, yielding large numbers of models trained on small amounts of data, reducing overall accuracy. Consequently, AlwaysGenerate, i.e. where $\alpha = 1$, underperforms threshold generate. We use ThresholdGenerate with cross-dataset validation based threshold computation in Fig. 8

ADM k value. For the ADM, we are selecting best-fit model where there is no obsolesence and the top k models when ModelDrift detects obsolesence and new submodels are still being generated. We vary k between 1 and 10, and show results for $k = (1, 5, 10)$ in Fig. 10. For lower values of k, accuracy has higher variance and fluctuates due to noisier signals from a single submodel that exhibits obsolescence. This is mitigated by using a team of submodels. For higher values of k, we include less confident submodels with more obsolete training data, reducing accuracy as well.

4.4 Limitations and Future Work

Novelty Detection. In ModelDrift, we used Lipschitz smoothness, and for distance metric, we used cosine similarity. Our future work will expand to more obsolesence detection metrics; for example, high density sets from (Jiang et al. 2018) are potential candidates for KO detection.

Submodel Generation. In submodel generation, we have assumed availability of labels from the FNC dataset. In practice, training data generation is time-consuming and expensive. Weak supervision methods such as Snorkel and EEWS (Rühling Cachay, Boecking, and Dubrawski 2021), combined with diverse knowledgebases (Dun et al. 2021) can be used for automatic training data generation.

ADM. We explored only empirical k values for the ADM. There can be technically grounded approaches as well, such as historical L values for each classifier. It is also possible to amortize the computation cost of ADM over a batch of predictions, improving submodel retrieval costs. The trade-offs between the computational costs of various design choices and their performance are interesting topics of future work.

5 Conclusion

KO Detection and Recovery. Using a large-scale COVID fake news data set over 25 months, we show strong evidence of new-normal and KO for representative static classifiers. We also present our ModelDrift metric for obsolesence detection, and evaluate it with respect to COVID-19 fake news detection. To demonstrate the feasibility of recovering from KO, we built a modular, adaptive classifier manager called Argus that incorporates ModelDrift and an adaptive model selector based on ModelDrift outputs. Argus achieves 2x higher accuracy compared to static classifiers, as well as 1.3x higher accuracy compared to fixed-window classifiers.

Ongoing and Future Work. We recognize the tremendous achievements of static ML classifiers through successful capture of timeless knowledge in fixed data sets. Compared to many thousands of papers based on static classifiers, including significant areas such as concept drift, outlier/anomaly detection, and fake news, we further acknowledge the modest initial steps of describing KO and recovery from KO in this paper. We hope the strong experimental evidence, plus the discussion on future work, can motivate and encourage a wider recognition of the growing impact and importance of new-normal and KO, since effective recovery from KO will be needed for sustained performance in new, evolving, and challenging applications such as COVID fake news detection.

References

Chen, E.; Lerman, K.; and Ferrara, E. 2020. Tracking social media discourse about the covid-19 pandemic: Development of a public coronavirus twitter data set. *JMIR Public Health and Surveillance*, 6(2): e19273.

Chen, M. F.; Fu, D. Y.; Adila, D.; Zhang, M.; Sala, F.; Fatahalian, K.; and Ré, C. 2022. Shoring Up the Foundations: Fusing Model Embeddings and Weak Supervision. *arXiv preprint arXiv:2203.13270*.

Dugas, A. F.; Jalalpour, M.; Gel, Y.; Levin, S.; Torcaso, F.; Igusa, T.; and Rothman, R. E. 2013. Influenza forecasting with Google flu trends. *PloS one*, 8(2): e56176.

Dun, Y.; Tu, K.; Chen, C.; Hou, C.; and Yuan, X. 2021. Kan: Knowledge-aware attention network for fake news detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 81–89.

Enders, A. M.; Uscinski, J. E.; Klofstad, C.; and Stoler, J. 2020. The different forms of COVID-19 misinformation and their consequences. *The Harvard Kennedy School Misinformation Review*.

Gama, J.; Žliobaitė, I.; Bifet, A.; Pechenizkiy, M.; and Bouchachia, A. 2014. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4): 1–37.

Gruppi, M.; Horne, B. D.; and Adalı, S. 2022. NELA-GT-2021: A Large Multi-Labelled News Dataset for The Study of Misinformation in News Articles. *arXiv preprint arXiv:2203.05659*.

Hussna, A. U.; Trisha, I. I.; Karim, M. S.; and Alam, M. G. R. 2021. COVID-19 Fake News Prediction On Social Media Data. In *2021 IEEE Region 10 Symposium (TENSYMP)*, 1–5. IEEE.

Jiang, H.; Kim, B.; Guan, M.; and Gupta, M. 2018. To trust or not to trust a classifier. *Advances in neural information processing systems*, 31.

Lazer, D.; Kennedy, R.; King, G.; and Vespignani, A. 2014. The parable of Google Flu: traps in big data analysis. *Science*, 343(6176): 1203–1205.

Li, Y.; Lee, K.; Kordzadeh, N.; Faber, B.; Fiddes, C.; Chen, E.; and Shu, K. 2021. Multi-Source Domain Adaptation with Weak Supervision for Early Fake News Detection. In *2021 IEEE International Conference on Big Data (Big Data)*, 668–676. IEEE.

Nakamura, K.; Levy, S.; and Wang, W. Y. 2019. r/fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection. *arXiv preprint arXiv:1911.03854*.

Pidhorskyi, S.; Almohsen, R.; and Doretto, G. 2018. Generative probabilistic novelty detection with adversarial autoencoders. *Advances in neural information processing systems*, 31.

Poursaeed, O.; Katsman, I.; Gao, B.; and Belongie, S. 2018. Generative adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4422–4431.

Pu, C.; Suprem, A.; Lima, R. A.; Musaev, A.; Wang, D.; Irani, D.; Webb, S.; and Ferreira, J. E. 2020. Beyond artificial reality: Finding and monitoring live events from social sensors. *ACM Transactions on Internet Technology (TOIT)*, 20(1): 1–21.

Rühling Cachay, S.; Boecking, B.; and Dubrawski, A. 2021. End-to-End Weak Supervision. *Advances in Neural Information Processing Systems*, 34.

Su, J.-C.; Tsai, Y.-H.; Sohn, K.; Liu, B.; Maji, S.; and Chandraker, M. 2020. Active adversarial domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 739–748.

Suprem, A.; Arulraj, J.; Pu, C.; and Ferreira, J. 2020. ODIN: Automated Drift Detection and Recovery in Video Analytics. *Proc. VLDB Endow.*, 13(12).

Suprem, A.; and Calton, P. 2022. FNC - A Large Scale Dataset for COVID-19 Fake News Detection. *CoRR*.

Suprem, A.; and Pu, C. 2022a. Evaluating Generalizability of Fine-Tuned Models for Fake News Detection.

Suprem, A.; and Pu, C. 2022b. MiDAS: Multi-integrated Domain Adaptive Supervision for Fake News Detection. *CoRR*.

Urner, R.; and Ben-David, S. 2013. Probabilistic lipschitzness a niceness assumption for deterministic labels. In *Learning Faster from Easy Data-Workshop@ NIPS*, volume 2, 1.

Žliobaitė, I.; Pechenizkiy, M.; and Gama, J. 2016. An overview of concept drift applications. *Big data analysis: new algorithms for a new society*, 91–114.