Ensemble and Incremental Learning for Norm Violation Detection

Thiago Freitas dos Santos[1,2] Nardine Osman[1] Marco Schorlemmer[1]
[1]Artificial Intelligence Research Institute, IIIA-CSIC, Barcelona, Catalonia, Spain
[2] Universitat Autònoma de Barcelona, Catalonia, Spain
{thiago,nardine,marco}@iiia.csic.es

ABSTRACT

The use of norms to guide and coordinate interactions has gained tremendous attention in the multiagent community. However, as the interest moves towards dynamic socio-technical systems, where human and software agents interact and interactions are required to adapt to the human's changing needs, new challenges arise. For instance, different agents (human or software) might not have the same understanding of what it means to violate a norm (e.g., what characterizes hate speech), or that their understanding of a norm might change over time (e.g., what constitutes an acceptable response time). The challenge is to address these issues by learning the meaning of a norm violation from the limited interaction data. For this, we propose a framework that learns and updates the meaning of a norm violation from interaction data through a combination of ensemble and incremental learning techniques. Ensemble learning handles the imbalance class distribution of the interaction stream, and incremental learning is used to continuously update the ensemble models as community members interact, which is an essential feature to keep the ensemble models in accordance with the latest community view on the meaning of norm violation. We evaluate the proposed approach in the use case of Wikipedia article edits, where interactions revolve around editing articles and the norm in question is prohibiting vandalism. Results show that the proposed framework can learn the meaning of a norm violation in a setting with data imbalance and concept drift.

KEYWORDS

Norm Violation; Concept Drift; Imbalanced Dataset; Ensemble Learning; Incremental Learning

ACM Reference Format:
Thiago Freitas dos Santos, Nardine Osman, and Marco Schorlemmer. 2022. Ensemble and Incremental Learning for Norm Violation Detection. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 22), Online, May 9–13, 2022*, IFAAMAS, 9 pages.

1 INTRODUCTION

The ability to continuously learn what constitutes norm violation, as understood by a given community, and detect when such violation happens is essential for any normative system that intends to regulate the behavior of its interacting agents (in this work, referred to as community members). This is specially critical when we consider that discrimination, hate speech and cyberbullying represent real damage to people's lives and interactions, besides affecting the community experience and engagement in online platforms [17, 24, 31]. Thus, the main challenge of this work is to learn what a community understands as norm violation by using examples of behaviors depicted as such. To do that, we are interested in finding and adapting the definition to norm violation as interactions unfold. This is not only important to online community domains, since elements present in norm violation are also of interest to fields in which detecting misbehave can prevent infractions (e.g., credit card frauds, personal information leakage and network infiltration).

Previous works in the realm of norms and normative systems have dealt with different challenges that arise in the field, with a series of proposals to handle mechanisms for norm conflict detection [1, 14], norm synthesis [26, 28] and norm emergence [23, 27, 35]. Besides, several domains have benefited from this field, applying the concepts of norms and normative systems to the prevention of discrimination by Machine Learning (ML) models [11], to the formalization of contracts and laws [16, 32], and to handle ethical dilemmas and moral values [2, 36]. In this work, we are particularly interested in supporting normative systems with mechanisms for learning from interactions and the feedback of agents (human or artificial) to help decide what is considered a norm violation.

Some interesting approaches to detect norm violation in online communities have been proposed, with applications to Wikipedia [4, 34, 40], Stack Overflow [9] and Reddit [6]. However, these approaches could not cope with the continuous update of the system used to classify an action as norm violation, and consequently they could not handle the evolution of the community view about what constitutes a norm violation. We argue that the understanding of a norm violation evolves over time (say, what is considered hate speech may change rapidly as new members are incorporated and interactions unfold), and it must follow the current view of the community. These limitations are addressed in the present work by proposing a framework that handles the interactions of an online community as a stream of actions with an imbalanced class distribution and the presence of concept drift. In other words, a stream of actions that contains more data describing regular behavior than data describing violation behavior, aggregating to that, changes in how the community members understand norm violation. Unlike existing approaches, community members' feedback are incorporated to update the understanding of norm violation.

To achieve this goal, we investigate the combination of ensemble and incremental learning in a framework that can learn the meaning of norm violation, adapt to changes in community view, and incorporate feedback from community members in the learning procedure. In our context, this combination offers the following advantages. First, it handles the imbalance of the dataset, which is particularly useful in cases with norm violation behavior, since

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

usually this kind of behavior happens less frequently than regular behavior. Second, it continuously updates the base classifiers of the ensemble, adding to the framework the ability to adapt to changes in the community view (concept drift). To do that, only the most recent data is used to learn the meaning of norm violation, discarding the need to treat and maintain past information. Third, it facilitates the incorporation of feedback from community members as the ground truth about what is a norm violation, which is aligned with our view that a system's understanding of norm violation needs to adapt to that of its users (in our case, community members).

The experiments (Section 4) describe the implementation of two incremental learning techniques, mini-batch learning and online learning. These are used to train the base classifiers of the ensemble. In this work, Feedforward Neural Networks are the models present in the ensemble. This implementation was evaluated in the use case of Wikipedia article edits. Results show that the proposed approaches can learn the meaning of norm violation in the context of an online community with imbalanced class distribution (only around 7% of the data described edits with vandalism) and in the presence of concept drift (changes in the community view). Here, an edit is described by the tuple (X, y), in which X is the set of features of an action and $y \in \{0, 1\}$ is its class label, 0 denotes regular behavior, while 1 denotes vandalism behavior.

The remainder of this paper is divided as follows. Section 2 presents the basic mechanisms used by our proposed framework, which is described in Section 3. Section 4 shows its application to the use case of Wikipedia article edits and Section 5 discusses the results. Related literature is presented in Section 6, and we give our conclusions and propose our future work in Section 7.

2 BACKGROUND

This section presents the base concepts upon which this work is built. First, we start by presenting the ensemble strategy to deal with the imbalanced nature of the dataset. Then, we describe the incremental learning approach used to continuously train the machine learning models present in the ensemble.

2.1 Ensemble Learning

Dealing with the detection of norm violating behavior usually leads to the case of imbalanced datasets. This happens because regular (or expected) behavior is more common than violations. Thus, solutions that deal with domains in these settings must be equipped with the ability to handle class distribution imbalance. Otherwise, the solutions tend to be biased towards the class that describes regular behavior. To tackle this issue, we use ensemble learning, which can be defined as the generation and combination of different ML models (e.g., neural networks, random forest and logistic regression) to solve a predictive task [33]. The main idea present in this technique is that by combining multiple ML models, using a voting scheme, the errors of a single model will be compensated by the others, thus the overall performance of the ensemble would be better than the performance of a single component [12].

There are different ensemble classification methods that can be used to build a classification system, Dong et al. [12] highlight some important ones, Bagging, AdaBoost, and Random Forest. Bagging is an interesting method to deal with the imbalanced dataset challenge investigated in this work. This technique finds a solution by training different base classifiers in different subsets of the initial dataset. Then, the ensemble uses majority voting to decide the final output. As an example, in a binary classification task with an imbalanced dataset DT, it is possible to divide DT into two subsets, majority class subset M and minority class subset P (the number of instances in these sets is represented by $|M|$ and $|P|$, respectively). In this context, the main goal is to train an ensemble E with N number of balanced datasets $B_{DT} = \{B_1, ..., B_N\}$. Each $B_i \in B_{DT}$ is a dataset with a similar class distribution, and $N = |M|/|P|$. In this manner, because the number of instances in $P \subseteq DT$ is smaller than number of instances in $M \subseteq DT$, subsets in B_{DT} have size $2 * |P|$ and are created with $|P|$ non-overlapping instances from M, while all instances of P are replicated to each subset.

The bagging method, as described above, can be applied to train ML models in an a offline or in a mini-batch manner. However, this method cannot be used in an online setting (in which training happens one instance at a time). To solve this issue, modifications to the bagging procedures are necessary. Thus, Wang et al. [38] present a resampling strategy to deal with imbalanced dataset for the online case. This strategy considers two approaches, the Oversampling-based Online Learning (WEOB1) and the Undersapmling-based Online Learning (WEOB2), with the addition of weight adjustment over time. WEOB1 and WEOB2 work to adjust the learning bias from M to P by resampling instances from these subsets. Specifically, oversampling increases the number of minority instances, while undersampling decreases the number of majority instances. Like the traditional bagging strategy, online bagging creates different classifiers and then trains each classifier $C \in E$ a K number of times by considering only the current data point. K is defined by the $Poisson(\lambda = 1)$ distribution. As data becomes available, the λ parameter is calculated dynamically according to the imbalance ratio. In this manner, if there is a new instance in P, then K is increased. But if there is a new instance in M, then K is decreased.

2.2 Incremental Learning

Since we are dealing with online communities, it is important to consider how data is made available. Usually, systems must work with a stream of data that arrives sequentially. In this context, there are different ways to build a system capable of solving the problem at hand. Techniques differ in how they handle the stream of data, and consequently, how the algorithms are trained. Following this idea, we can separate training techniques in two big groups: offline learning and incremental learning.

Offline learning deals with the complete dataset, and in this case it is not possible to update the trained model. To incorporate new knowledge, a full train process from the beginning is necessary [18], which is the main drawback of this approach when we must handle non-stationary domains. Besides, maintaining and treating all the data for this kind of learning can be costly and complex (specially when we consider data regulations specified by different entities and legislators) [20].

On the other hand, incremental learning is the approach that deals with the drawbacks present in offline learning. To do that, this technique incrementally learn as new data is made available. This is particularly interesting in online communities, since the ML model must be constantly updated as people interact with each other and a change in understanding emerges. In this work, we are concerned with mini-batch and online learning. Mini-batch learning creates and uses small sets of the data that arrives to continuously train ML models. Since we only deal with the most recent instances that compose the present data block of a fixed size, the process is neither as costly nor as complex as offline learning [20, 21]. Online learning can be seen as a special case of mini-batch learning, in which the batch size is 1. Thus, as soon as data is made available, it is possible to update the ML model, discarding the need to store this data point, and consequently avoiding the complexities of data treatment.

By continuously updating the base ML model as data is made available, incremental learning approaches are an interesting way to investigate problems in which there is the presence of concept drift, which in our work can be defined as the change in view of the community members about what is regular behavior and what is violation behavior. It is possible to identify the change in community behavior by observing the joint distribution $P_t(X, y)$ over time [22, 37], where $x \in X$ is a feature value, $y \in \{0, 1\}$ is the associated class label that denotes regular or norm violation behavior, and t the current time stamp. Then, to compare two moments in time and detect a possible concept drift, we refer to the following: $P_t(X, y) \neq P_u(X, y)$, where u is a time stamp in the past. Gama et al. [15] define three ways to categorize concept drift: change in the prior probability of classes $p(y)$, which is a change in the ratio between vandalism and regular behavior; change in the class conditional probabilities $p(X|y)$, which is a change in how vandalism and regular behavior are defined; this can affect the posterior probabilities of classes $p(y|X)$, which is a change in what the community understands as norm violation and regular behavior. This kind of data leads to what is referred to as real concept drift, which is the type of concept drift that interests us in this work.

3 THE ENSEMBLE INCREMENTAL LEARNING FRAMEWORK

In this section, we present the proposal of our work, a framework capable of learning the meaning of a norm violation by combining ensemble and incremental learning. The main idea is to deploy this framework in a normative system to support the enforcement of norms, especially when dealing with prohibited behavior.

Dealing with norm violation behavior usually leads to working with datasets that are imbalanced by nature, since violation behavior does not happen as often as regular (or expected) behavior. Thus, when building a solution to tackle learning in this setting, it is necessary to work with an approach capable of handling imbalance in class distribution. In this work, we investigate the use of ensemble machine learning to tackle this issue. Besides, we also apply two different approaches to continuously update the base ML models, 1) mini-batch, in which the learning algorithm is trained using blocks of data; and 2) online, in which the learning algorithm is trained using a single data point as soon as it is made available.

3.1 Mini-Batch Learning

As data is made available in a sequential manner, the algorithm starts to build blocks of data with fixed size N. As soon as a data block contains N data points, the algorithm is ready to start the training procedure of the ML classifiers. The mini-batch approach explored in this work (Algorithm 1) builds on top of two incremental ensemble algorithms, the Accuracy Updated Ensemble (AUE2) [5] and the Dynamic Updated Ensemble (DUE) [21]. The differences introduced by our approach are: 1) incorporating feedback to emphasize data points that had their class labels changed by the community; 2) using a replication-based oversampling technique that randomly replicates minority class instances present in the current data block, instead of using the SMOTE [8] oversampling technique that creates synthetic minority class samples. Besides, we also define a new metric (number of classifiers) to define the oversampling ratio for the minority instances (Algorithm 1, line 10).

```
1  Algorithm: Mini-Batch Training
2  Input: Current data block ($B_t$), set of majority instances
     ($M_t$), set of minority instances ($P_t$), set of instances with
     feedback ($F_t$), max number of classifiers ($MC$), max change
     in distribution ($CD$), and number of epochs ($NE$);
3  Output: Trained ensemble ($E$).
4  Initialize ensemble size. $ES \longleftarrow 0$
5  Initialize last imbalance ratio change. $IRC \longleftarrow 0$
6  while data block is available do
7      Pre-process $B_t$, no past data is used
8      Compute current imbalance ratio. $IR_t \longleftarrow |P_t|/|M_t|$
9      Compute current best ensemble size. $ES_t \longleftarrow |M_t|/|P_t|$
10     if $ES_t > MC$ then
11         Oversample minority class instances $\in P_t$
12         Update $ES_t$ with the new value for $|P_t|$
13     end
14     if $IRC = 0$ or $IR_t/IRC < 1 - CD$ then
15         Compute number of new classifiers.
             $NC \longleftarrow ES_t - ES_{t-1}$
16         $IRC \longleftarrow IR_t$
17     end
18     Emphasize set $F_t$ by oversampling with ratio $|P_t|/|F_t|$
19     Add $F_t$ to the data block $B_t$
20     for i=1; i<=$ES_t$; i++ do
21         Get a subset $SM_{t,i}$ from $M_t$, where $|SM_{t,i}|$=$|P_t|$ and
             $SM_{t,i} \cap SM_{t,u} = \emptyset$ $(u = 1, 2, ...i - 1)$
22         Create balanced dataset. $BD_{t,i} = SM_{t,i} \cap P_t \cap F_t$
23     end
24     Train $ES_t$ classifiers with $BD_t$ for $NE$ epochs;
25 end
```

Algorithm 1: The Mini-Batch Training procedure.

In our case, the majority class M represents expected behavior, while the minority class P represents norm violation behavior. Since we define an action as a set of features, we represent a data point with the tuple (X, y), in which X is the set of features of an action and $y \in \{0, 1\}$ is its class label. Thus, a data block is defined as B_t = $\{(X, y)_1, ..., (X, y)_N\}$, with N being the batch size. After the data is pre-processed, the algorithm starts by calculating the imbalance ratio (Algorithm 1, line 8) between sets P_t and M_t in the current data block B_t. Besides, set M_t and set P_t are used to calculate the number of classifiers in the ensemble ES_t (Algorithm 1, line 9).

To illustrate in more detail how Algorithm 1 works, it is interesting to use an example. Let us say that initially $t = 1$, $ES_t = 10$, and the imbalance ratio $IR_t = 0.07$. Then, after some time, at time step $t = 5$, a concept drift is noted, with IR_t changing to 0.03 and ES_t changing to 12. Next, if $ES_t > MC$, the algorithm oversamples set P_t by duplicating all minority instances (Algorithm 1, line 11), which prompts the update of the best ensemble size. The algorithm then checks if the imbalance ratio has changed by some pre-defined factor CD (it is worth to mention that the community members may decide on an appropriate number for this value). After that, the algorithm incorporates community feedback (Algorithm 1, line 19) in order to present to the training procedure, relevant data about change in the community view. Then, ES_t balanced datasets are created from data block B_t. Each balanced dataset is composed of non-overlapping data points from M_t, all data points from P_t and all feedback data points from F_t (Algorithm 1, line 22). Lastly, the algorithm executes the training procedure for each of the ES_t ML base classifiers with the balanced datasets $\in BD_t$.

3.2 Online Learning

```
1  Algorithm: Online Training
2  Input: Current data point ($D_t$), data point feedback ($F_t$),
     desired class distribution ($DD$), sampling rate ($SR$), max
     change in class distribution ($CD$);
3  Output: Trained ensemble ($E$).
4  Initialize ensemble of classifiers. $E \longleftarrow \{NeuralNetworks\}$
5  Initialize last imbalance ratio change. $IRC \longleftarrow 0$
6  while data point is available do
7      Pre-process $D_t$, with running statistical values
8      Update partial class distribution $IR_t$
9      Update number of data points $N$
10     if $IRC > 0$ and $IR_t/IRC < 1 - CD$ then
11         Update desired distribution. Minority class increases
             by the ratio $IR_t/IRC$
12     end
13     Compute rate to draw from distribution.
         $R \longleftarrow SR * DD/(IR_t/N)$.
14     for Classifier $C \in E$ do
15         Calculate resample ratio. $RR \longleftarrow poisson(R)$
16         Train $C$ with the oversample data point
17     end
18     if $D_t \in F_t$ then
19         Oversample data point $F_t$ by duplicating
20         Train $E$ with $F_t$
21     end
22 end
```

Algorithm 2: The Online Training procedure.

Algorithm 2 describes the procedure to train the ensemble of classifiers in an online manner, which is built on top of the concepts described by Wang et al. [38] and Montiel et al. [25]. The first step is to create the ensemble of classifiers E (Algorithm 2, line 4). The number of base classifiers in E can be defined by the community, from expert knowledge or through initial experiments. For each data point D_t that is made available (i.e., for each action in an online community), the algorithm pre-processes D_t using the running statistical values. We are interested in the mean and the sum of squares, since these are used to normalize the incoming data point.

Different from the mini-batch approach, in online learning as soon as a single data point is made available, the training procedure is executed. However, this characteristic leads to a different way to calculate statistical values for the pre-process phase. In this case, the algorithm must compute running statistical values, which are updated at each time step and are less exact than the values computed using blocks of data [25]. To compute these values, the following equations are used:

$$RM_t \longleftarrow RM_{t-1} + ((V_t - RM_{t-1})/N_t) \qquad (1)$$

where RM_t is the updated running mean at time t for each feature that describes an action, RM_{t-1} is the last running mean, V_t is the new feature value, and N_t is the number of data points encountered until the current time t. With the running mean, it is possible to calculate the running sum of squares (SQ_t):

$$SQ_t \longleftarrow SQ_{t-1} + (V_t - RM_{t-1}) * (V_t - RM_t) \qquad (2)$$

Since it is not possible to know the data distribution for the complete dataset in online training, it becomes necessary to decide as interactions happen which portions of the data are going to be used for training. To tackle this, the algorithm checks for concept drift by calculating the change in the imbalance ratio IR (Algorithm 2, line 10). If the change is bigger than a defined threshold value, then the desired distribution DD is updated, which works to emphasize the minority class instances.

After updating DD, the algorithm calculates the rate in which to draw a random value (Algorithm 2, line 13) following the Poisson distribution. This value is used to determine the sampling strategy (oversampling or undersampling). For each classifier $C \in E$, the algorithm uses the Poisson distribution to determine how many times a data point is replicated for training [38]. Thus, the larger the imbalance ratio, the larger the number of times that minority data points are used for training. Although we use the work in [38] to calculate the resampling rate, future work will investigate the effect of applying alternative strategies to calculate this value [13].

Lastly, if the data point receives feedback from the community (represented by F_t), then F_t is oversampled to emphasize the provided information. Then, all classifiers in the ensemble are trained with F_t (Algorithm 2, line 20). Empirical experiments showed that oversampling presents a higher recall performance than the weighting scheme proposed by the other approaches.

4 EXPERIMENTS

This section describes how the incremental approaches presented earlier were applied to the use case of Wikipedia article edits. This use case is relevant because Wikipedia is an open and collaborative community, with a set of norms to maintain and organize its content [29]. Due to these characteristics, people with diverse backgrounds interact with each other, and therefore misunderstandings about what constitutes a norm violation might emerge. The dataset used in this work is provided by Wikipedia and it was annotated by Amazon's Mechanical Turk. Santos et al. [34] describe a taxonomy for this dataset, with details about the features that represent an action and the relationship between them.

To evaluate the performance of the proposed approaches, we design two different experiments:

- ***Learn the meaning of norm violation with no concept drift:*** In this case, the goal is to evaluate if the proposed algorithms can learn the meaning of norm violation. The data set contains 32.439 edits, with 2.394 vandalism edits (around 7%) and 30.045 regular edits (around 93%). The dataset is highly imbalanced. We use 10-fold cross validation to evaluate the performance. Classification recall is the chosen metric.
- ***Learn the meaning of norm violation with concept drift:*** In this case, the aim is to evaluate if the proposed algorithms can learn the meaning of norm violation in the presence of concept drift. To do that, we start by separating the complete dataset CD into two subsets IT and FT. IT contains data used to initially train the ensemble, with 1.197 vandalism edits and 15.022 regular edits; and FT contains data that incorporates the concept drift, with 1.197 vandalism edits and 15.023 regular edits. This separation is necessary because we want to demonstrate the algorithms' ability to incrementally adapt to new concepts. Thus, we start by training the algorithms with the subset IT. Only when the algorithms process all data points in IT, we start learning from the changing dataset FT. In this experiment, we are particularly interested in adding concept drift by changing what edits are labelled as vandalism (swap of class label). Since we do not have real feedback from community members, we simulate it by changing the dataset as follows: using only the vandalism subset $V_{FT} \in FT$, we apply the K-Means clustering algorithm to generate subgroups that contain data points most similar between themselves [19]. From this process, we obtain 4 subgroups, $G = \{0\text{: } 618,\ 1\text{: } 442,\ 2\text{: } 117,\ 3\text{: } 20\}$. The idea of getting these groups with similar data points is to keep the dataset consistent when simulating community feedback. For this experiment, we swap the class label from all data points $\in G_0$. Then, the class distribution changes, resulting in 15.641 regular edits and 579 vandalism edits. Consequently, the imbalanced ratio changes as well to $IR = 0.037$.

We build the ensemble using the Keras library [10]. Feedforward Neural Networks (FNN) are the base classifiers. To provide comparison between mini-batch and online learning, the FNN architecture is the same in both cases. Stochastic Gradient Descent (SGD), with a learning rate equal to 0.01, is used as the optimizer and the loss function is the Cross Entropy. The experiments were run on a 2.6GHz Intel Core i7-9750 with 16GB of RAM.

It is necessary to set specific parameters for the learning algorithms. In mini-batch learning, batch size is 512 and the number of epochs is 200. In online learning, the initial ensemble size is set to 12, the desired distribution is 50% for each class label (regular and vandalism), and the sampling rate is equal to 1. These values are found empirically and can affect the performance of the classifiers.

5 RESULTS AND DISCUSSION

This section presents the results of applying the algorithms considered in this work to the case of vandalism article edits in Wikipedia.

Figure 1 presents the graph that describes the overall recall score for the algorithms when applied to the first experiment (no concept drift). The learning curve for both approaches are similar and the Wilcoxon Signed-Rank Test (Table 2) attest this similarity. The null hypothesis was not rejected, thus there is no statistically significant difference between mini-batch and online learning. Although these algorithms are similar to the overall cases, the algorithms present a difference when specifically dealing with vandalism instances.

With the values described in Table 1 and the learning curve in Figure 2, it is possible to conclude that the mini-batch algorithm outperforms the online algorithm in the task of correctly classifying vandalism edits. Besides, it is also possible to verify the instability properties present in the online case. This is a characteristic of the algorithm caused by the training approach (since it considers only one point at a time) and by the resampling strategy used [21, 38, 39].

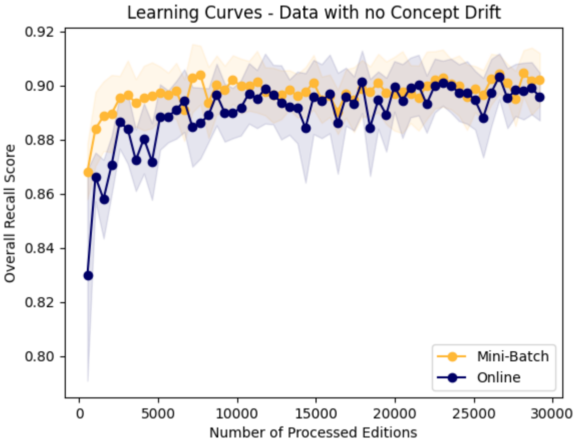

Figure 1: Overall Recall for the Mini-Batch and Online cases with no concept drift

Regarding the second experiment, Figure 3 shows the overall recall for the case with concept drift. During most part of the training procedure (until around 12.000 processed instances), mini-batch performs significantly better. The online learning algorithm presents higher variation and instability when drift is first introduced, taking more time for the performance to improve and to reach the same level of the mini-batch approach. Thus, online learning needs to process more data points to stabilize its learning performance. Towards the end of the training procedure, both approaches have similar overall performance, with no significant difference (Table 2).

In the case of classifying vandalism edits, the mini-batch approach significantly outperforms the online approach (Table 2). Figure 4 present the learning curve for vandalism classification. As in other cases, the online algorithm is more unstable, suffering with a significant drop in performance as the concept drift is first introduced. The comparison between the metrics for the overall and vandalism cases is important, mostly when we are dealing with a

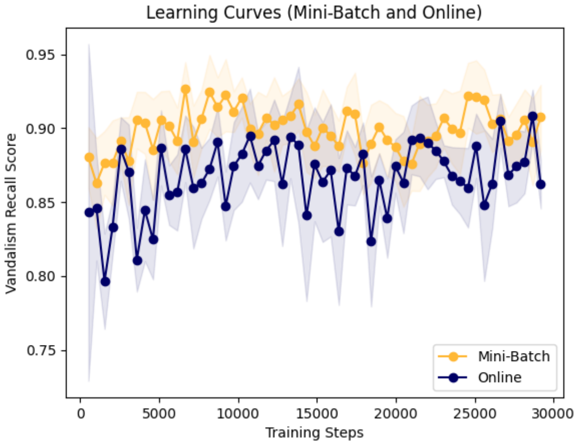

Figure 2: Vandalism Recall for the Mini-Batch and Online cases with no concept drift

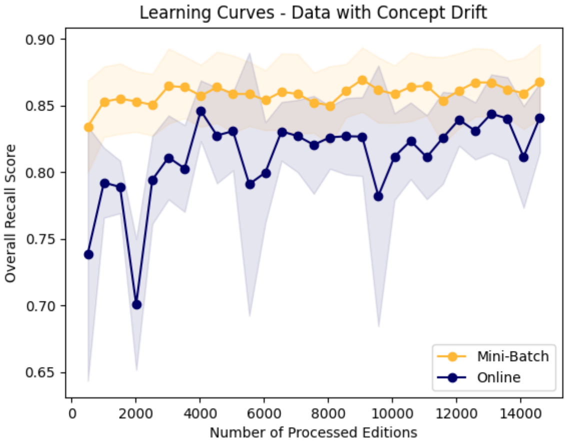

Figure 3: Overall Recall for the Mini-Batch and Online cases in the presence of concept drift.

highly imbalance dataset. If we do not consider how the algorithms specifically perform for a certain class, then we might evaluate wrongly the proposed approaches. Online learning presents a bias towards classifying the majority class, which improves the overall recall. Thus, the evaluation of the algorithms for specific labels is crucial to really understand their performance.

Figure 5 presents the recall specifically for the data that suffered the swap of class label (which we will refer to as the Re-label dataset). As the community starts to give feedback, the performance of the framework drops, which is expected, since we are introducing new information to the dataset. Then, as more data is made available and the ML models are incrementally trained, the ensemble is capable of learning that certain article edits should not be classified as vandalism anymore, thus adapting to the new view of the community. Table 2 shows that the online learning algorithm presents a significantly better performance in classifying this new community view (although the instability properties of the algorithm are also present in this case). The bias towards the majority class influences the performance of the online algorithm for this case, since by changing the classification label from vandalism to regular behavior, we increase the imbalance ratio.

To summarize, Table 1 presents performance information of each approach in the considered datasets and the time required for the training procedure, showing that the mini-batch learning is more efficient than online learning. While Table 2 describes the results for the Wilcoxon Signed-Rank Test, which compares the performance of the proposed approaches. The null hypothesis is that the samples were drawn from the same distribution, and the critical value $\alpha = 0.05$. Results show that the mini-batch approach is more suitable to classify vandalism edits, offering a more stable performance and adapting more quickly to concept drift. While the online approach presents a bias towards the majority class and consequently, in our concept drift case, a bias towards the changed data. Besides, this approach suffers a significant drop in performance when classifying vandalism edits. Here, we note the need to further investigate and explore the effects of different imbalance strategies and of the incorporation of community feedback on the algorithm performance, since the online approach can learn the new concept, but at the cost of the performance in the minority class.

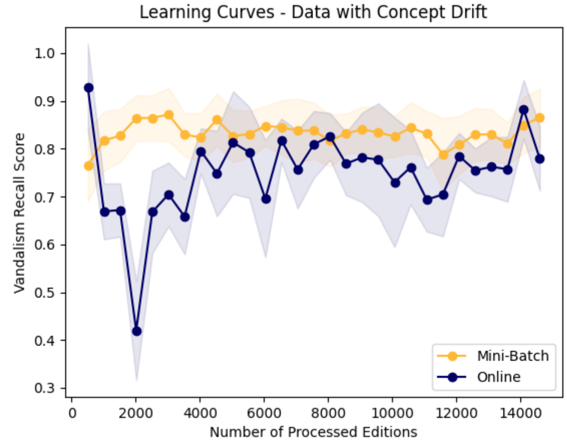

Figure 4: Vandalism Recall for the Mini-Batch and Online cases in the presence of concept drift.

Finally, it is possible to conclude that both approaches are suitable to deal with the challenge of learning the meaning of norm violation in the context of an online community. Mini-batch offers more stability, better performance at vandalism detection and faster training, since it needs to process a smaller number of instances to solve a task. On the other hand, online learning offers the flexibility of updating the model as soon as data is made available, with no need to maintain and create data blocks, while keeping an acceptable classification performance. Thus, the choice of approach must take into consideration the requirements of the community.

Table 1: Summary of the performance results of Mini-Batch and Online Learning applied to the Wikipedia article edits dataset. Three settings are considered: 1) dataset with no concept drift (Original); 2) dataset with concept drift, swap of class label; and 3) dataset with only the data that suffered the change (Re-Label). Training time in seconds to process 512 editions (batch size).

Dataset	Method	Recall±Std	Regular Recall±Std	Vandalism Recall±Std	Training Time±Std
Original	Mini-Batch	**0.9023**±0.0097	0.8971±0.0091	**0.9075**±0.0219	**4.0947**±0.7032
	Online	0.8959±0.0088	**0.9297**±0.0094	0.8622±0.0164	10.4159±0.9021
Concept Drift	Mini-Batch	**0.8679**±0.0280	0.87085±0.0120	**0.8651**±0.0597	X
	Online	0.8408±0.0259	**0.9025**±0.0319	0.7792±0.0674	X
Re-Label	Mini-Batch	0.8708±0.0120	X	X	X
	Online	**0.9277**±0.0284	X	X	X

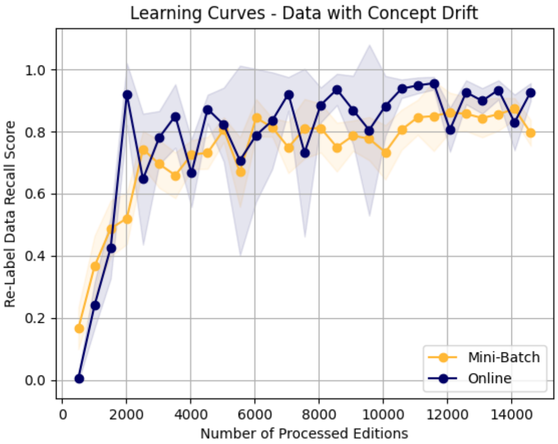

Figure 5: Re-Label Recall for the Mini-Batch and Online cases, vandalism edits re-labeled to regular edits.

Table 2: Summarized comparison between the recall performance of mini-batch and online learning. The Wilcoxon Signed-Rank Test is used to obtain the P-values.

Dataset	P-values		
	Overall	Regular	Vandalism
Original	0.2754	0.0039	0.0058
Concept Drift	0.1308	0.0273	0.0371
Re-Label	0.0019	X	X

6 LITERATURE REVIEW

This section presents related work to the research reported in this paper. Specifically, we cite literature focusing on detecting detrimental behavior in online communities by using machine learning. The idea is to present different approaches that aim to deal with this issue in different communities, which highlights the importance of research in the field. For example, Risch and Krestel [31] describe several deep learning approaches to deal with toxic comments. In [3], the authors use Natural Language Processing (NLP), machine learning and feature representation techniques as the basis to build a solution that handles hate speech. Chandrika et al. [7] report and compare the application of several machine learning algorithms to the task of detecting abusive comments online, with Neural Network presenting better results than other approaches. Additionally, we also focus on works that deal with incremental learning in an environment with concept drift and imbalanced dataset [15, 22, 30]. Gama et al. [15] and Lu et al. [22] present surveys on concept drift, with different applications to solve this challenge in several domains, while Ren et al. [3] build an ensemble to deal with imbalanced dataset and concept drift using a sampling strategy that considers previous seen data to enhance the current minority set.

Other research also focused on the Wikipedia online community to detect norm violation. Anand and Eswari [4] apply Deep Learning to classify a comment as abusive or not, based on a dataset from the talk page edit. Santos et al. [34] use Logistic Model Tree to learn the meaning of norm violation, and they provide a taxonomy that describes the relationship between the features of an action. However, these differ from our research in the sense that they do not cope with concept drift, and consequently do not incorporate community feedback to update their models.

One work that also explores machine learning to detect norm violation is presented in [9], which explores norm violation on the Stack Overflow (SO) community. As in our work, Cheriyan et al. [9] use specific data about the context of the SO community to train the ML models. In this case, instead of article edits, Cheriyan et al. [9] analyze comments that were posted on the site. The violation is defined by the presence of hate speech and abusive language. The main difference in our work is our focus on the application of an incremental learning approach to continuously update the ML models responsible for detecting when norm violation occurs, while Cheriyan et al. [9] focus on the use of a recommendation system to detect and recommend, to the community members performing the action, alternatives to the comment they are posting.

Using an approach that applies ensemble learning to help in the task of comment moderation in Reddit, Chandrasekaran et al. [6] created a system that uses the concept of cross-community learning to train different ML models on different data (provided by several communities), namely the Crossmod approach. The goal is to detect a violation in a specific community by understanding how other communities would classify a certain comment. Different from our approach that uses ensemble learning to create ML models with balanced portions of the dataset, Crossmod collects information from different communities to train the ensemble of classifiers.

Regarding the use of incremental learning in a setting with class distribution that is highly imbalanced, the work by Lebichot et al. [20] builds a solution capable of detecting credit/debit card frauds.

Like our use case, these transactions have a sequential nature, are highly imbalanced and present concept drift. The proposed approach in [20] reports better results than the traditional offline learning approaches. To enhance incremental learning, Lebichot et al. [20] use ensemble learning to reduce variance and improve stability, while transfer learning is used to deal with information that was learned in a different task. One difference in our approach is that we use an active process to detect concept drift, which is better suited to deal with major changes that happen in time. Lebichot et al. [20], on the other hand, apply a passive strategy to concept drift, since in their domain, several concept drifts happen daily. Another major difference is the way to deal with an imbalanced dataset. While we use ensemble, their work uses parameter tunning of a dense neural network model. In this case, the models that compose the ensemble are independently trained and the final output is the average of the probability scores.

Zeng et al. [21] present an incremental learning approach that focuses on emphasizing misclassified instances in the update procedure of the models that compose the ensemble (DUE). Another interesting characteristic of DUE is that it keeps a limited number of classifiers in the ensemble, this is done to ensure efficiency. As in our work, DUE uses an ensemble to handle data imbalance, with no need to access past data. The oversampling technique used in [21] is the SMOTE, while we oversample by duplication. One major difference between our approaches is the inclusion of a feedback component that uses data provided by the community to emphasize instances with a swap of class label.

Zhang et al. [41] present an ensemble framework to handle concept drift in an imbalanced dataset context, the Resample-based Ensemble Framework for Drifting Imbalance Stream (RE-DI). This approach makes use of a resampling buffer to keep instances of the minority class to handle the class distribution over time. Besides, members of the ensemble that perform worst on the minority class receive less weight. RE-DI maintains a long-term static classifier, to handle gradual change, and a set of dynamic classifiers, to handle sudden concept drift, which only focus on recently received data. In the case of the dynamic classifiers, their weights are incrementally decreased as time goes by and they are dynamically created and replaced. The goal is that by the end of the training, the last concepts were learned by the classifiers. Different from our approach, in which we use oversampling to emphasize the minority class and undersampling to decrease the influence of the majority class in the training procedure, RE-DI uses a buffer (making use of past data), in which last added data is more used than the older ones.

7 CONCLUSION AND FUTURE WORK

This work has proposed mechanisms that support normative systems with learning from interactions and feedback of agents (human or artificial) to help decide what is considered a norm violation. We provide the foundations for working with norms whose meaning can change over time, like what is considered hate speech or acceptable response time. The proposed mechanisms build on ensemble and incremental learning. We focus on two challenges that emerge in these domains: 1) the imbalanced nature of the dataset; and 2) the adaptation to the changing community view on the meaning of norm violation. Thus, the main contribution is the addition of feedback data (in future this data shall be collect from a real online community) to update the machine learning model as interactions unfold. A dataset from Wikipedia article edits was used to investigate norm violation. This dataset describes a binary classification problem, with each action being labeled as regular or vandalism behavior, with an imbalance class distribution, in which only 7% of the data represent vandalism behavior.

Experiments were conducted on two different dataset configurations. First, we evaluated the algorithms in the case with no concept drift, focusing on learning the meaning of norm violation. The recall metric for the vandalism, regular and overall cases were explored. The second experiment was designed to evaluate the algorithms in a context with concept drift, specifically drift that occurs due to the swap of class labels. For this experiment, we highlighted the need of feedback from community members as a manner to enhance the proposed approaches. Since we did not have real feedback from the community, we used a simulation strategy, in which we created different subgroups of the vandalism dataset and changed the label. This simulation assumes that the feedback is consistent, since we are grouping similar editions together, thus our interpretation of the results naturally comes from this consistency.

Results show that both proposed approaches are suitable for detecting norm violation in an online community. For the first experiment, both approaches reported acceptable results, although mini-batch significantly outperformed online learning in the detection of vandalism edits. As for the second experiment, the mini-batch approach had more stability in the learning process and better performance in classifying vandalism actions, while online learning presented a significant drop in performance for the vandalism classification, due to the bias towards the majority class. Considering that, future work will focus on investigating different resampling strategies to deal with this performance decrease.

As we argue that feedback from community members can provide information on how a community understands norm violation, future work shall focus on getting real feedback. This is not only interesting because of feedback collection, but also from the point of view of how the community members will agree on the definition of norm violation. Besides, for the ensemble of classifiers to decide if an action is norm violation, we are investigating the adoption of different strategies, from a simple voting scheme (used in this paper) to something more complex as deliberation.

Lastly, it is not only important to detect when a norm violation occurs, but also to understand the reasons behind such detection. Thus, in future work, we shall focus on the interpretation of the ML models. This kind of information would enhance the community experience, since our system would be capable of explaining to the user what the community understands as non-acceptable behavior. We also note here that, to provide this information, it is assumed that violation happens due to unintentional behavior.

ACKNOWLEDGMENTS

This research is supported by the WeNet EU-funded project (grant agreement No 823783), the Rhymas Spanish-funded project (PID2020-113594RB-100), and the *Ajuts a grups de recerca consolidats* from the Generalitat de Catalunya (grant no. 2017 SGR 172).

REFERENCES

[1] João Paulo Aires and Felipe Meneguzzi. 2021. Norm Conflict Identification Using a Convolutional Neural Network. In *Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XIII*, Andrea Aler Tubella, Stephen Cranefield, Christopher Frantz, Felipe Meneguzzi, and Wamberto Vasconcelos (Eds.). Springer International Publishing, Cham, 3–19.

[2] Nirav Ajmeri, Hui Guo, Pradeep K Murukannaiah, and Munindar P Singh. 2020. Elessar: Ethics in Norm-Aware Agents.. In *AAMAS*. 16–24.

[3] Areej Al-Hassan and Hmood Al-Dossari. 2019. Detection of hate speech in social networks: a survey on multilingual corpus. In *6th International Conference on Computer Science and Information Technology*.

[4] M. Anand and R. Eswari. 2019. Classification of Abusive Comments in Social Media using Deep Learning. In *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*. 974–977.

[5] Dariusz Brzezinski and Jerzy Stefanowski. 2014. Reacting to Different Types of Concept Drift: The Accuracy Updated Ensemble Algorithm. *IEEE Transactions on Neural Networks and Learning Systems* 25, 1 (2014), 81–94.

[6] Eshwar Chandrasekharan, Chaitrali Gandhi, Matthew Wortley Mustelier, and Eric Gilbert. 2019. Crossmod: A Cross-Community Learning-Based System to Assist Reddit Moderators. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW (Nov. 2019), 30.

[7] C. P. Chandrika and Jagadish S. Kallimani. 2020. Classification of Abusive Comments Using Various Machine Learning Algorithms. In *Cognitive Informatics and Soft Computing*, Pradeep Kumar Mallick, Valentina Emilia Balas, Akash Kumar Bhoi, and Gyoo-Soo Chae (Eds.). Springer Singapore, Singapore, 255–262.

[8] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[9] Jithin Cheriyan, Bastin Tony Roy Savarimuthu, and Stephen Cranefield. 2017. Norm violation in online communities–A study of Stack Overflow comments. In *Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XIII*. Springer, 20–34.

[10] François Chollet et al. 2015. Keras. https://keras.io.

[11] Natalia Criado, Xavier Ferrer, and Jose M Such. 2020. A normative approach to attest digital discrimination. *arXiv preprint arXiv:2007.07092* (2020).

[12] Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. 2020. A survey on ensemble learning. *Frontiers of Computer Science* 14, 2 (2020), 241–258.

[13] Hongle Du, Yan Zhang, Ke Gang, Lin Zhang, and Yeh-Cheng Chen. 2021. Online ensemble learning algorithm for imbalanced data stream. *Applied Soft Computing* 107 (2021), 107378. https://doi.org/10.1016/j.asoc.2021.107378

[14] Stephen Fenech, Gordon J Pace, and Gerardo Schneider. 2009. Automatic conflict detection on contracts. In *International Colloquium on Theoretical Aspects of Computing*. Springer, 200–214.

[15] João Gama, Indrundefined Žliobaitundefined, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A Survey on Concept Drift Adaptation. *ACM Comput. Surv.* 46, 4, Article 44 (March 2014), 37 pages.

[16] Xibin Gao and Munindar P Singh. 2014. Extracting normative relationships from business contracts. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. 101–108.

[17] Kishonna L. Gray. 2018. Gaming out online: Black lesbian identity development and community building in Xbox Live. *Journal of Lesbian Studies* 22, 3 (2018), 282–296. PMID: 29166214.

[18] Steven C.H. Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. 2021. Online learning: A comprehensive survey. *Neurocomputing* 459 (2021), 249–289. https://doi.org/10.1016/j.neucom.2021.04.112

[19] K Krishna and M Narasimha Murty. 1999. Genetic K-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 29, 3 (1999), 433–439.

[20] Bertrand Lebichot, Gian Marco Paldino, W Siblini, L He-Guelton, F Oblé, and G Bontempi. 2021. Incremental learning strategies for credit cards fraud detection. *International Journal of Data Science and Analytics* (2021).

[21] Zeng Li, Wenchao Huang, Yan Xiong, Siqi Ren, and Tuanfei Zhu. 2020. Incremental learning imbalanced data streams with concept drift: The dynamic updated ensemble algorithm. *Knowledge-Based Systems* 195 (2020), 105694.

[22] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. 2018. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering* 31, 12 (2018), 2346–2363.

[23] Samhar Mahmoud, Nathan Griffiths, Jeroen Keppens, and Michael Luck. 2012. Efficient norm emergence through experiential dynamic punishment. In *ECAI 2012*. IOS Press, 576–581.

[24] Lavinia McLean and Mark D Griffiths. 2019. Female gamers' experience of online harassment and social support in online gaming: a qualitative study. *International Journal of Mental Health and Addiction* 17, 4 (2019), 970–994.

[25] Jacob Montiel, Max Halford, Saulo Martiello Mastelini, Geoffrey Bolmier, Raphael Sourty, Robin Vaysse, Adil Zouitine, Heitor Murilo Gomes, Jesse Read, Talel Abdessalem, and Albert Bifet. 2021. River: machine learning for streaming data in Python. *Journal of Machine Learning Research* 22, 110 (2021), 1–8.

[26] Javier Morales, Michael Wooldridge, Juan A Rodríguez-Aguilar, and Maite López-Sánchez. 2018. Off-line synthesis of evolutionarily stable normative systems. *Autonomous agents and multi-agent systems* 32, 5 (2018), 635–671.

[27] Andreasa Morris-Martin, Marina De Vos, and Julian Padget. 2019. Norm emergence in multiagent systems: a viewpoint paper. *Autonomous Agents and Multi-Agent Systems* 33, 6 (2019), 706–749.

[28] Ronen Nir, Alexander Shleyfman, and Erez Karpas. 2020. Automated synthesis of social laws in strips. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 9941–9948.

[29] Martin Potthast and T. Holfeld. 2010. Overview of the 1st International Competition on Wikipedia Vandalism Detection. In *CLEF*.

[30] Siqi Ren, Bo Liao, Wen Zhu, Zeng Li, Wei Liu, and Keqin Li. 2018. The gradual resampling ensemble for mining imbalanced data streams with concept drift. *Neurocomputing* 286 (2018), 150–166.

[31] Julian Risch and Ralf Krestel. 2020. Toxic comment detection in online discussions. In *Deep Learning-Based Approaches for Sentiment Analysis*. Springer, 85–109.

[32] Paolo Rosso, Santiago Correa, and Davide Buscaldi. 2011. Passage retrieval in legal texts. *The Journal of Logic and Algebraic Programming* 80, 3-5 (2011), 139–153.

[33] Omer Sagi and Lior Rokach. 2018. Ensemble learning: A survey. *WIREs Data Mining and Knowledge Discovery* 8, 4 (2018), e1249.

[34] Thiago Freitas dos Santos, Nardine Osman, and Marco Schorlemmer. 2021. Learning for Detecting Norm Violation in Online Communities. *arXiv preprint arXiv:2104.14911* (2021).

[35] Bastin Tony Roy Savarimuthu, Maryam Purvis, Martin Purvis, and Stephen Cranefield. 2008. Social norm emergence in virtual agent societies. In *International Workshop on Declarative Agent Languages and Technologies*. Springer, 18–28.

[36] Marc Serramia, Maite Lopez-Sanchez, and Juan A Rodriguez-Aguilar. 2020. A qualitative approach to composing value-aligned norm systems. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. 1233–1241.

[37] Heng Wang and Zubin Abraham. 2015. Concept drift detection for streaming data. In *2015 international joint conference on neural networks (IJCNN)*. IEEE, 1–9.

[38] Shuo Wang, Leandro L. Minku, and Xin Yao. 2015. Resampling-Based Ensemble Methods for Online Class Imbalance Learning. *IEEE Transactions on Knowledge and Data Engineering* 27, 5 (2015), 1356–1368.

[39] Shuo Wang, Leandro L Minku, and Xin Yao. 2018. A systematic study of online class imbalance learning with concept drift. *IEEE transactions on neural networks and learning systems* 29, 10 (2018), 4802–4821.

[40] Andrew G West and Insup Lee. 2011. Multilingual Vandalism Detection Using Language-Independent & Ex Post Facto Evidence. In *CLEF Notebooks*.

[41] Hang Zhang, Weike Liu, Shuo Wang, Jicheng Shan, and Qingbao Liu. 2019. Resample-Based Ensemble Framework for Drifting Imbalanced Data Streams. *IEEE Access* 7 (2019), 65103–65115. https://doi.org/10.1109/ACCESS.2019.2914725