Int. J. Human–Computer Studies 160 (2022) 102772

Contents lists available at ScienceDirect

International Journal of Human - Computer Studies

journal homepage: www.elsevier.com/locate/ijhcs

Modeling and mitigating human annotation errors to design efficient stream processing systems with human-in-the-loop machine learning

Rahul Pandey [*,a], Hemant Purohit [a], Carlos Castillo [b,c], Valerie L. Shalin [d]

[a] George Mason University, 4400 University Dr, Fairfax, VA, USA
[b] Universitat Pompeu Fabra, Plaza de la Merc, Barcelona, 10-12, Spain
[c] ICREA, Pg. Lluís Companys 23, Barcelona, Spain
[d] Wright State University, 3640 Colonel Glenn Hwy, Dayton, OH, USA

ARTICLE INFO

Keywords:
Human-centered computing
Active learning
Annotation schedule
Memory decay
Human-AI collaboration

2020 MSC:
00-01
99-00

ABSTRACT

High-quality human annotations are necessary for creating effective machine learning-driven stream processing systems. We study hybrid stream processing systems based on a Human-In-The-Loop Machine Learning (HITL-ML) paradigm, in which one or many human annotators and an automatic classifier (trained at least partially by the human annotators) label an incoming stream of instances. This is typical of many near-real-time social media analytics and web applications, including annotating social media posts during emergencies by digital volunteer groups. From a practical perspective, low-quality human annotations result in wrong labels for retraining automated classifiers and indirectly contribute to the creation of inaccurate classifiers.

Considering human annotation as a psychological process allows us to address these limitations. We show that human annotation quality is dependent on the ordering of instances shown to annotators and can be improved by local changes in the instance sequence/order provided to the annotators, yielding a more accurate annotation of the stream. We adapt a theoretically-motivated human error framework of mistakes and slips for the human annotation task to study the effect of ordering instances (i.e., an "annotation schedule"). Further, we propose an error-avoidance approach to the active learning paradigm for stream processing applications robust to these likely human errors (in the form of slips) when deciding a human annotation schedule. We support the human error framework using crowdsourcing experiments and evaluate the proposed algorithm against standard baselines for active learning via extensive experimentation on classification tasks of filtering relevant social media posts during natural disasters.

According to these experiments, considering the order in which data instances are presented to a human annotator leads to increased accuracy for machine learning and awareness of the potential properties of human memory for the class concept, which may affect annotation for automated classifiers. Our results allow the design of hybrid stream processing systems based on the HITL-ML paradigm, which requires the same amount of human annotations, but that has fewer human annotation errors. Automated systems that help reduce human annotation errors could benefit several web stream processing applications, including social media analytics and news filtering.

1. Introduction

Filtering high-volume, high-velocity data streams is a typical process in many application domains such as journalism, public health, and crisis management. In this process, an avalanche of data must be filtered and classified to prevent recipient information overload and filter failure (Shirky, 2008). These continuous streams of data are often noisy, sparse, and redundant. Humans cannot keep pace with the high velocity and volume of data. A purely human-annotation based filtering system does not scale. These data streams are also problematic for purely automated/machine-annotation based filtering systems; depending on the application, they may have limited accuracy. In the case of supervised classifiers for such automated filtering, data sampled from previously collected streams can bootstrap classifier training. However, it is

* Corresponding author.
E-mail address: rpandey4@gmu.edu (R. Pandey).
URL: http://mason.gmu.edu/~rpandey4/ (R. Pandey).

https://doi.org/10.1016/j.ijhcs.2022.102772
Received 17 June 2020; Received in revised form 22 December 2021; Accepted 4 January 2022
Available online 10 January 2022

invaluable to have annotations on samples specifically from the current data stream to adapt the pre-trained classifier model for the new data. Hence, to achieve high accuracy in this process, online human annotation tasks are needed within an active learning paradigm (Gama et al., 2014), sometimes at a large scale. Fortunately, social media and mobile devices have provided an unprecedented opportunity for the public to participate by volunteering in stream processing applications for digital humanitarianism, citizen science purposes, etc. A popular option for annotating complex data streams has been to create hybrid stream processing systems through a composition of human annotation tasks and automatic online classification (Imran et al., 2013; Lofi and Maarry, 2014).

In this paper, we study a hybrid online classification setting that categorizes relevant instances from a social media data stream using human annotation tasks and an active learning paradigm. Drawing on both classic (Ebbinghaus, 1913) and contemporary (Anderson, 2000) cognitive psychology, we analyze the effective decay related to attentional processes (described below) in human memory in contributing to errors while doing human annotation tasks.[1]

Data challenges in hybrid stream processing. A key challenge in stream processing is temporal variation in the concept space. This includes changes in the distribution of data that leads to change in decision boundaries (virtual drifts), changes in the population from which future samples will be drawn (population drift), and changes in the definition of a concept (concept drift) (Gama et al., 2014) as illustrated in Fig. 1. For example, consider the task of processing crisis-related instances posted on social media during a natural disaster, such as a hurricane. To find instances that can help emergency managers in a response agency, we need to categorize them as irrelevant or relevant for actionable services (Purohit et al., 2018b), and in the case of relevant instances, further categorize them into fine-grained information classes such as infrastructure damage, donations, and so on (Castillo, 2016). In this setting, both virtual drifts and population drifts occur as the crisis unfolds. An example of virtual drift is content variation as a crisis evolves (Olteanu et al., 2015; Sutton et al., 2015). Consider a class concept such as *caution and advice*. In the beginning, instances might be urgent and generic, warning the public about a potentially dangerous event (such as a hurricane warning). Later, the same category of instances may become more specific and less urgent (such as warning people to avoid drinking contaminated water). An example of population drift is change in the prevalence of different class instances, which follow a certain progression across many events (Olteanu et al., 2015). For instance, immediately after a sudden onset crisis event, instances of caution and advice appear. Later on, other classes of information may be prevalent such as appeals for relief donations. These temporal variations are expected. They have a potential effect on annotation quality due to the learning behavior of human annotators about the representation of a class concept, which, in turn, impacts the entire system when used to train the automatic part of a hybrid system.

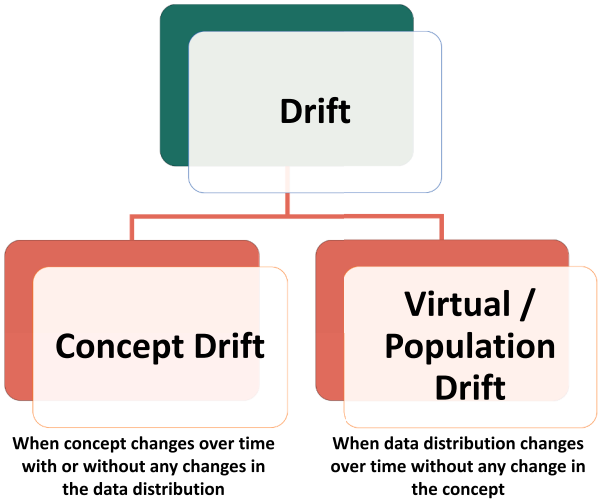

Fig. 1. Categories of drift in streaming data Gama et al. (2014).

Human challenges in hybrid stream processing. Human factors in the annotation process affect the quality of annotations for hybrid stream processing systems. Systems that rely on some form of crowdsourcing are affected by cognitive properties of human annotators, including their attentional heuristics (e.g., the fit with prior experience, the associated positive or negative affect) and vigilance (the ability to sustain high attention over time) (Burghardt et al., 2018). High mental workload (e.g., demands on inference and decision making) causes a deterioration in annotation quality, known as *annotator burnout* (Marshall and Shipman, 2013), which can cause increased fatigue and reduced motivation to maintain accuracy. To prevent annotator burnout, one may cap the maximum number of annotation tasks per unit of time that the annotator must perform, which can reduce workload (Purohit et al., 2018a). Nevertheless, human error persists in the execution of annotation tasks.

Psychologists distinguish between two types of human error: mistakes and slips (Reason, 2000). Mistakes result from incorrect or incomplete knowledge (Reason, 2000). In the annotation task context, this corresponds to annotators who have not yet grasped the concept to be annotated or who are annotating new instances for which they have not yet acquired a correct representation. Slips are errors in the presence of correct and complete knowledge (Norman, 1981; Reason, 2000), i.e., annotator knowledge is correct, but idiosyncrasies in the activation of this knowledge modify accessibility, resulting in an incorrect annotation assignment. Persistent slips after a large number of examples may result from vigilance decrements in underlying attentional processes (Wiener, 1987). The classic serial position effect (Murdock, 1962) supports this distinction between knowledge-based mechanisms and attentional processes, in which early items are properly encoded and hence remembered while later items are only stored temporarily and subject to decay. Item order matters, particularly when the content to be acquired changes over time (Jacoby et al., 2001), as explained above under data challenges.

1.1. Contributions

This paper extends our prior conference publication (Pandey et al., 2019), with the following new contributions.

- First, we present a generic human error framework of mistakes and slips, adapted from the psychological theories that cover some common types of human errors and apply it to study human errors possible in an annotation task for streaming data, using the active learning system in a HITL-ML paradigm (Sections 3 and 4).
- Second, we extend the validation of the proposed human error framework using a quantitative error model by presenting details of both lab-based and crowdsourcing-based testing experiments for the annotation task to filter relevant information from social media data streams collected during crises (Sections 5 and 6).
- Third, we present a novel method for human error-mitigation in the active learning paradigm for designing a stream processing system against several baselines (Section 7). We also provide additional novel insights on different automated algorithmic approaches to prevent human error (Section 8).

[1] We appreciate the distinction between absent memory traces and the challenges of retrieval (Tulving and Pearlstone, 1966). For the purposes of this paper, the net result is memory decay that results in effective forgetting.

The application of the proposed human error framework can be used to design Human-AI collaboration strategies and improve the performance of a human-in-the-loop approach for hybrid stream processing systems.

2. Background

2.1. Online active learning

To the best of our knowledge, existing types of online active learning methods focus only on the possible machine/algorithmic errors. Prior literature (e.g., Almeida et al., 2018; Gama et al., 2014) provides extensive surveys of the different active learning paradigm-based methods. The primary categories include one group focused on a better sampling of the instance space for querying (e.g., addressing concept drift Žliobaitė et al., 2014), and another group focused on better learning of a discriminatory model.

To improve sampling of the instance space, prior research has explored different mechanisms to drop the outdated/drifted class instances. The simplest way is to consider a fixed window over instance sequence and sample past instances from that window as they arrive. Windows can be specified by size and sampling on a first come first serve basis, or by time and sampling of instances from the last *t* seconds/minutes/hours. These approaches do not represent well the characteristics of a data stream. Hence, alternative approaches were utilized in the past that uniformly sample and therefore, retain the characteristics of the underlying incoming stream of instances (Delany et al., 2005; Ng and Dash, 2008; Salganicoff, 1993; Vitter, 1985; Yao et al., 2012; Zhao et al., 2011; Žliobaitė, 2011). Other work does not completely drop all past instances but instead, reduce their weights for updating the classifier by an age-dependent factor (Helmbold and Long, 1994; Klinkenberg, 2004; Koren, 2010; Koychev, 2000; 2002).

To improve acquisition of the discriminatory model, prior research has mainly explored two strategies. The first is called the *blind adaptation strategy*, which retrains the model without any detection of changes (Klinkenberg and Joachims, 2000; Klinkenberg and Renz, 1998; Lanquillon, 2001; Widmer and Kubat, 1996). The other way of improving learning includes an informed strategy, which updates the model whenever a certain criterion is fulfilled like change detectors (Bifet and Gavaldá, 2006; Hulten et al., 2001). These criteria can also be aligned with the adaptation strategy (Gama et al., 2006; Ikonomovska et al., 2011), called model-integrated detectors.

Our research premise is that the improvement of both types of the above active learning methods for stream processing systems require consideration of potential human annotation errors during the querying process as well, to be efficient and accurate in predictive model learning for the classifier. For simplicity, our method builds upon the blind adaptation strategy, which updates the model as we sample the instances in a sequence-based window.

2.2. Human annotation task and psychological processes

Annotation quality can be affected by many factors. At the most basic level, a human annotation task can be conceptualized using signal detection theory (SDT) and its two fundamentally distinct parameters of discriminability (*d'*) and decision criterion bias (*beta*). Discriminability concerns the relationship between the mean signal strength of the distributions of positive and negative class instances. Nearly overlapping distributions pose difficult discrimination, such as using photographs to distinguish older from younger individuals that are close in age, whereas the overlap in signals is much smaller for gender discrimination from photographs (Nguyen et al., 2014). *beta* in signal detection theory is an independent parameter, concerning the position of the decision criterion on these overlapping distributions, dropping it down to be more liberal to reduce the chance of misses (false negatives) or moving it up to be more conservative to reduce the chance of false alarms (false positives). The classic manipulation of *beta* is achieved by the imbalanced distribution of positive and negative class instances or weighing the cost of misses and false alarms differently.

Signal detection theory has been applied to the analysis of sequential industrial inspection tasks, resulting in the supposition of a vigilance decrements that affects judgment quality over time (Mackworth, 1948; Wiener, 1987). This classic approach fails to recognize change over time in the relevant features in the data. Moreover, though influential in the perception literature, signal detection theory also fails to address several issues that arise in conceptual judgment tasks. Much later, Kahneman and Tversky (1979) elaborated a theory of bias to describe incoherence in decision making depending upon the influence of contexts such as prior belief or loss aversion. In this sense *bias* is an umbrella term to characterize the systematic departure of decisions from rational analysis, which can account for a human annotator's errors given a drifting data stream.

The annotation task is typically multi-class for a variety of applications that adds task complexity and hence, cognitive demand on the annotator. Following an initial training period, the failure to attain agreement between annotators on a multi-class coding scheme, known as inter-rater reliability in the social sciences (Creswell and Poth, 2016) has been generally attributed to a flawed coding scheme, rather than the cognitive challenge of learning the scheme and systematically applying it over time.

Similarly, for information scientists developing machine learning models for data analytics, the appreciation of annotation as a psychological process emerges from the requirement for annotating large training datasets over an extended period of time, where each judgment matters. Although human annotation is often regarded as a gold standard, information scientists have noted that class imbalance leads to difficulties in appropriately representing the minority class to help human annotators learn the class concepts (Bröder and Malejka, 2017; Grant et al., 2017). Information scientists have also observed that annotation styles affect human annotation quality to factors such as objectivity and descriptiveness (Cheng and Cosley, 2013). Furthermore, annotation expertise affects quality, particularly in difficult tasks (Hansen et al., 2013). Item position with respect to its class concepts (referred to as "annotation schedule"), cognitive demand, and attentional processes may lead to annotation error (Burghardt et al., 2018). Missing from both theory and method for human annotation tasks is a framework to organize and investigate specific human error types in the annotation tasks of hybrid stream processing systems. Moreover, unlike purely psychological research, the erroneous annotation of an individual item has consequences for the machine learning model, which learns to automate the data annotation process.

3. Human error framework

Our focus on a human error framework is intended to reveal the different human reasoning processes that result in erroneous annotations. We assume a preliminary phase of the annotation task where instruction provides an initial understanding. However, this preliminary phase results in a mental representation of the concept (e.g., infrastructure damage during a disaster) at the beginning of an extended annotation task that is only partial, in the sense that the changing boundaries and nuances about a concept are learned while the annotations are performed. We also assume that the annotator can develop a mental representation of a concept by seeing a sufficient number of examples of this concept, even in the typical case where the examples are not annotated a priori.

Following Reason (2000)'s human error taxonomy built on Norman (1981)'s theory and broadly applied, including the analysis of medical domain errors (Zhang et al., 2004), we distinguish two classes of errors for the human annotation task: mistakes and slips. Mistakes result from the absence of a correct cognitive representation of a concept. Slips are errors that happen despite acquiring the correct cognitive representation

of a concept. Based on these broad classes, we present a framework of human errors in the annotation task for stream processing in Table 1 and explain the main error types below. We do not claim that all classes of human errors are equally prevalent or are equally consequential.

3.1. Serial ordering-induced mistakes

The annotation schedule in which the tasks are presented to an annotator may prevent the annotator from adequately apprehending a concept, hence introducing mistakes. The main types of mistake include:

- Concept not acquired yet: The annotator is asked to annotate an instance of a class for which s/he has not seen a sufficient number of examples to learn the concept overall.
- Erroneous concept with missing or extraneous features: At best this blends categories and at worst creates uncertainty.

3.2. Serial ordering-induced slips

The annotation schedule in which judgments occur may cause slips, in which the annotator erroneously annotates an instance even if s/he has a correct representation of its concept. We identify two main cases for the types of slips:

- Slip favoring an available activated concept: In this case, metacognitive monitoring (vigilance) is suspended, resulting in an instance label that comes easily to his/her mind. Serial position, particularly the persisting activation of recent judgments, especially when reinforced with repetition has the potential to exacerbate slips that result in a false alarm (false positive).
- Slip ignoring a minimally available concept: The complement of activation is effective inhibition. In this case, the correct category does not come to the annotator's mind, because its activation is too small compared to other categories. The annotator has not forgotten the concept, but it is inaccessible, resulting in the application of the available label instead of the correct one. This results in a miss (false negative).

Because slips result from activation failures of fundamentally correct knowledge, concept training is unlikely to help. Both cases result from extreme local divergence from the base rate or the loss of metacognitive function, boredom, or fatigue. These can be addressed with proper annotation schedules.

Both types of errors described above, induced by primarily serial ordering constraints, are particularly vulnerable to a classification scheme that changes over time and underlying processes of proactive and retroactive inhibition on knowledge acquisition. As a whole, mistakes can be reduced by ordering instances to facilitate learning. This includes both a sufficient number of examples of each concept presented and reminders from old concepts, so that the annotator reinforces persistent and emergent critical distinctions between classes. Because the observable behavior (erroneous classification) is the same for both slips and mistakes, but the mitigation is different, the technical challenge is to identify the mechanism behind the observed error.

Table 1
Framework of human annotation errors in hybrid stream processing applications. [*empirically studied in this article].

Type of error	Potential cause	Mitigation approach
**Mistakes induced by serial ordering*	• Concept not acquired yet	• Show frequent concept examples for learning, potentially informed by judicious selection such as near misses
**Slips induced by serial ordering*	• Imbalanced presence of a high-availability concept or a low-availability concept	• Limit extreme divergence from base rate for concept instances
Mistakes and slips due to temporal and environmental constraints	• Concept memory decayed due to oversight in rapidly finishing the annotation task	• workload and stress of the external environment causing vigilance challenges in learning a concept
	• Intervene reminders for concept examples	• Limit the number of concepts to annotate or the number of instances in a time unit

3.3. Other influences (temporal and environmental constraints) that induce mistakes and slips

As described in the background section, time and environmental constraints such as workload and its resulting stress during the annotation task can also cause human error. These constraints can cause vigilance and oversight challenges to the human annotators, causing slips and potentially, mistakes due to insufficient attention spent on the example instances to learn the concept. To limit the scope for the first study on such human annotation framework for stream processing applications, we do not consider such constraints in the experimentation and plan to explore these in future work. One of the future explorations to address such constraints include providing work specification, an amount of work, and a working environment that is appropriate, providing pauses to the worker, and so on.

In the following sections, we present three different experimental frameworks to reason about the existence of human errors and their mitigation by an algorithm: lab-based, crowdsourcing-based, and simulation-based. The lab-based error testing framework is similar to the conventional approach to experimentation in psychology, with greater control over the annotation task environment; however, a lab-based framework is difficult to scale to multiple annotators. The crowdsourcing-based approach can help to remedy the scalability challenge of the experimental setup. However, it provides less control on the setup to capture the annotators' behavior and their unacquired knowledge. Lastly, the simulation-based approach allows us to generate the streaming data samples, emulate human errors through an automated agent (referred 'oracle'), and demonstrate error mitigation techniques. However, it may oversimplify observations of the real world and thus, could not capture the behaviors of all the different human annotators out there.

4. Annotation task for hybrid stream processing systems

A hybrid stream processing application requires human annotation to adapt and improve the classification model continuously with new annotated instances. We define the specific annotation task for human error testing and mitigation to classify an instance from a given sequence/stream of Twitter instances (tweets) into k classes.

We use labeled datasets from prior work in crisis informatics that contain labeled tweets related to natural disasters (Alam et al., 2018). We re-crawled the tweet instances from Twitter's API for acquiring metadata such as timestamp and discarded any tweets deleted since the data were originally collected. The three natural disasters include major natural hazards affecting Central and North America in 2017 "Hurricane Harvey, Hurricane Maria, and Hurricane Irma. The labels were created using a crowdsourcing platform, classifying instances into four major categories:

- *infrastructure and utility damage (c1):* information about any physical damage to infrastructure or utilities
- *rescue, volunteering, and donation effort (c2):* information about offering help through volunteering efforts by a community of users
- *affected individuals (c3):* information about the condition of the individuals during this disaster event

- *not relevant or cannot judge (c4):* instance either does not contain any informative content or hard to decide.

We considered human labels with a confidence score (computed by the crowdsourcing platform for agreement between multiple annotators (Alam et al., 2018)) greater than 65% for ground truth labeled instances in our experimentation.

5. Lab-scale annotation error testing

5.1. Overview

We focus on verifying the effect of decayed memory behavior (Ebbinghaus, 1913), which underlies the above-mentioned error types of serial ordering-induced mistakes & slips and impacts the performance of both human annotation and ML models in the hybrid stream processing system.

5.1.1. Memory decay curve

Psychologists have been studying memory-decay behavior in the context of learning and acquiring new knowledge for more than a century. The Ebbinghaus Curve "shown in Fig. 2" is a fundamental and enduring contribution to the study of human memory. We observe from Fig. 2 an exponential decay of memory retention as the time since first learning passes. This exponential behavior has been widely observed in the psychology literature (Brown, 1958; Melton, 1963; Peterson and Peterson, 1959). Moreover, Loftus (1985) and Anderson and Schooler (1991) have used an exponential function with respect to time to model memory decay or retention. Hence, inspired by the Ebbinghaus curve, we model the *decaying_score* for the memory retention of an annotator for a particular class (c). Specifically, we model the *decaying_score* for c by observing how the annotator correctly annotates instances as an exponential function over time t_c lapsed over its last seen annotated instance. We define the $decaying\_score(c)$ function in Eq. (1) below:

$$decaying\_score(c) \propto e^{-t_c} \tag{1}$$

Moreover, to compute the probability of human annotators making an error, we use a function that is a vertical reflection of the aforementioned $decaying\_score(c)$ function along the x-axis. For our experiments, we assume a parameterized *sigmoid* function to compute the annotation error probability given the similar asymptotic nature of the vertical reflection of the exponential memory decay curves. We define the *error_probability_score* function in Eq. (2) below:

$$error\_probability\_score(c) = \gamma \times \frac{1}{1 + e^{-\alpha t_c + \lambda}} \tag{2}$$

Here the parameters α, λ, and γ represent different memory decaying intensities of human annotators. As each human annotator has individual memory retention capability, the intensity of making errors in annotations varies for different human annotators, and hence these parameters help mimic different human memory decay behavior. For verifying the above function for human memory decay, we conducted a small-scale controlled lab study.

5.2. Participants

We selected three students working as Graduate Research Assistants at an Information Technology research lab on social media research to volunteer in this study. The participants included one female and two male students (authors refrained from participation), and all of them were in the age range of 25–30. These students have been working with social media mining for more than six months. They have routinely participated in categorizing social media messages in the past. Hence, they were well acquainted with social media messages during emergencies and were given a brief training session and clear instructions on annotating different class instances.

5.3. Design

The experiment used an annotation system for the annotation task defined in Section 4. The input was a sequence of tweet instances for the Hurricane Harvey disaster. This synthetic input sequence contained instances with a random amount of irrelevant instances (noise) between ground truth annotated instances of any class to better observe human memory decay of the class and resulting errors. We added between one to four irrelevant instances (randomly selected) between each of the ground truth annotated instances, and they were marked as "not_relevant_or_cant_judge". Our data stream contained 800 instances.

5.4. Procedure

All annotators were asked to annotate the instances into four class labels. Three annotators labeled a given instance in the stream separately, with no ability to backtrack.

5.5. Method

Once we collected the three annotators' responses, we observed whether or not the annotators reveal memory decay effects for that class. Given the correct class concept for a given instance, we first store the time difference since we last observe any previous instance of that class concept. Moreover, we store the number of annotators who identified the correct class concept of that given instance. We plot the number of annotators who correctly identified the class concepts and the time difference for every instance.

5.6. Results

Fig. 3 shows the plot of how many instances of each class were correctly identified with respect to the time difference (in steps) between the appearance of consecutive instances of that class in the data stream. The size of the circle represents the number of instances that 'y' annotators have correctly annotated with 'x' time difference since they last observed that class instance in Fig. 3. Due to the sequential nature of the experiment, most of the class instances appear in very few (< 10) steps, and hence, the figure is left-skewed. Moreover, we observe that many annotators incorrectly annotate the instances despite the class instances appearing frequently. This shows that multiple other influences can cause the annotators to make errors as described in Section

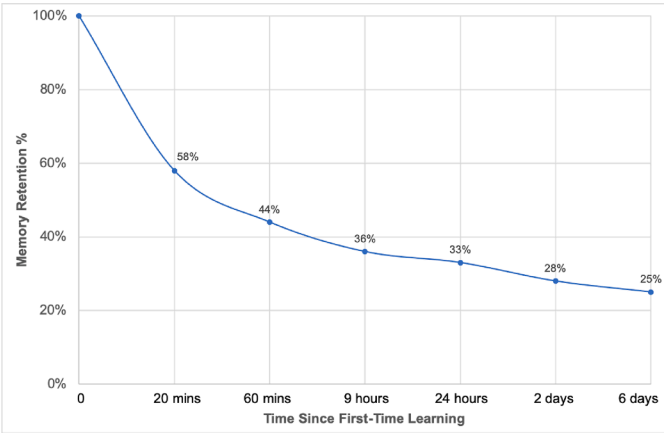

Fig. 2. The effect of memory decay studied in Psychology (Ebbinghaus, 1913) over time in learning or retaining conceptual knowledge. We investigate such effects of memory decay on the human annotation quality for hybrid stream processing systems and corresponding mitigation approaches.

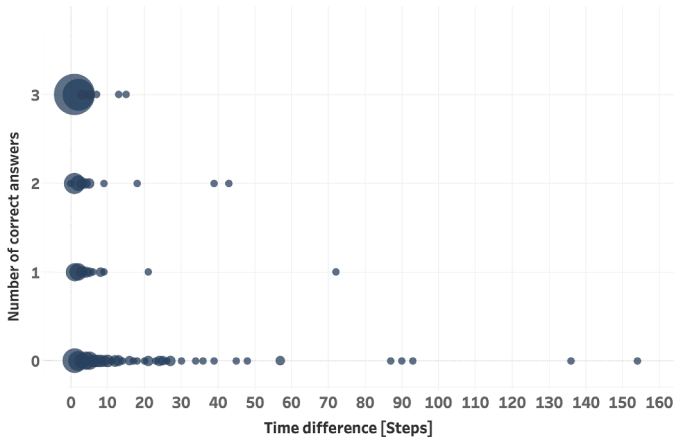

Fig. 3. The temporal distribution of correct answers by human annotators shows a similar pattern as Fig. 2 for memory decay behavior. The probability of incorrect answers (error) increases for annotating a given instance in the sequence with increased steps between instances of the same concept.

3. However, we also observe that when the time difference between the class concepts increases, the chance of the annotators correctly identifying the class decreases exponentially and finally tends to zero correct annotation. In comparison, the highest chance of all the annotators picking the correct class in annotating an instance is when the time difference is close to zero.

5.7. Discussion

These results support the quantitative model of memory decay behavior as described in Section 5.1.1, verifying the exponential nature of memory decay (ref. Eq. (1)) of annotators as they last see the instance of a particular class. Hence, we use an exponential function to compute the memory decay score for each class in our proposed *Error-Avoidance Sampling* technique for error mitigation described in Section 7.1.3. Moreover, as discussed in Section 5.1.1, a parameterized *sigmoid*-based error probability function from Eq. (2) can be used to induce an error in our algorithmic simulation experiments, later on, mimicking the real-world environment for a human annotation task in the stream processing systems. We understand that the number of instances with large time steps was low. Hence, increasing the number of instances and the number of annotators would have shown more explicit exponential behavior of memory decay. Further, we observe a high inter-rater agreement (0.82 Cohen's Kappa score) between two of the three annotators, which is higher in comparison with the similar social media annotation task in the literature (0.68 Cohen's Kappa score from Zhou et al., 2021), but the third annotator had low inter-rater agreements with the other two (0.48 and 0.46 Cohen's Kappa score respectively). We also observe that with the increase in time difference, none of the annotators could correctly identify a class. Within the limitation of the scalability of a lab-scale study, we still achieve the same exponential decay behavior widely studied and suggested by the past psychology literature (Anderson and Schooler, 1991; Loftus, 1985). Furthermore, our proposed error testing and mitigation approach can use any function, which closely resembles the vertical reflection properties of the exponential function and not just the *sigmoid* function as an error-inducing function.

6. Crowd-scale annotation error testing

6.1. Overview

Our crowdsourcing-based experiments seek to measure the prevalence of both mistakes and slips, and the conditions under which these appear. The goal of these crowd-scale experiments is to motivate the design of algorithms seeking to minimize these errors.

6.2. Participants

We asked ten human judges to annotate six fixed sequences of instances per schedule for two schedules using the crowdsourcing platform.

6.3. Design

For the crowd-scale experiment, we generated two types of annotation schedules for the task described in Section 4, corresponding to mistakes and slips. For practical reasons of the cost and time of crowdsourcing, we limited the length of the schedules to 20 instances. For constructing the schedules, we used the labeled data as ground truth, and based on the labeled data distribution, we chose the minority class c3 (instances about "affected individuals") as our target class for error analysis. The selection of c3 as the target class is used as an example to create a different annotation schedule because it was appearing the least in the data distributions, and hence, more prone to error.

For studying slips induced by serial ordering, we examine the case when instances of a target class (c3) are positioned with a mix of short and long gaps in the annotation schedule.

We assume that non-uniform and infrequent occurrences cause the annotators to deactivate the knowledge of the target concept class, potentially leading to memory decay behavior. Thus, we hypothesize that the annotation error per position of the target class instance should increase at the end of the annotation schedule (H1). Similarly, we study mistakes induced by serial ordering when instances of a target class (c3) are positioned with equal gaps in an annotation schedule. We observe the annotation error at each position in the schedule where an instance of the target class appears. We permute the instances of the target class on these positions. We hypothesize that uniform and frequent occurrences would allow the annotators to acquire the knowledge of the target class slowly. Thus, we hypothesize that the annotation error per position of the target class should reduce as we move toward the end of the annotation schedule (H2).

The first annotation schedule corresponding to slip errors (H1) is {c4, c1, c2, **c3**, c1, **c3**, c4, c1, c4, c1, c4, c2, c1, c4, c1, c2, c4, c2, c4, **c3**} and the second schedule corresponding to mistake errors (H2) is {c4, c1, c2, c1, c4, c2, c1, c4, **c3**, c1, c2, c4, **c3**, c1, c2, c1, **c3**, c2, c4, c4}. The underlined class label indicates the occurrence of target class instances and their position for analysis. For the three positions of the target class (c3) in an annotation schedule, we permuted the instances shown in those positions, leading to six experimental cases for each type of schedule.

6.4. Procedure

We used the Figure Eight platform (now called Appen) to obtain annotations for each type of experimental case. Initially, each participant was given a set of example tweet messages for each class label to train them for the annotation. Next, they were asked to annotate 20 instances into the four labels described in Section 4. For simplicity of the labeling process, we separated the fourth label described in Section 4 into "not relevant" and "cannot judge" respectively. We specified that the users should only look at the text while picking the label and not open any external link. The compensation amount for each task was $0.15. Figure Eight platform uses several layers of protection to prevent inattentive workers. It encourages workers to maintain a high reputation within the system and removes anomalous workers, including automated responses ("bots"). They also indicated whether the response was tainted or not in any form, and we only included non-tainted responses. After filtering all the tainted responses, we extracted a total of 120 responses for the two schedules. The reason for collecting many responses

from the crowdsourcing platform is to avoid the chances of other kinds of error influences such as inattentiveness, workload, and stress to overpower the cause of serial-ordering-based errors due to our proposed schedules.

6.5. Method

With the Figure Eight platform, we could not enforce participants to participate in only one experiment. Therefore, we filtered out the responses from participants who have participated in multiple experiments. As a result, after filtering, we got 44 responses for the slips-based annotation schedule and 38 responses for the mistakes-based annotation schedule, respectively. Once we filtered out the responses from duplicate participants (who participated in multiple experiments), we analyzed the responses on both types of annotation schedules. We extracted annotations of the target positions for each response and checked if the annotators had incorrectly labeled to any class label other than **c3**. If incorrectly labeled, we considered the error score as one, else zero. We accumulated error scores from all the responses made at first, second, and third positions, respectively. Next, we took the union of the errors made at the first and second positions and compared this with the third position's error using statistical significance testing.

6.6. Results

Table 2 shows the results of crowdsourcing for the micro-average error rate (average error by an annotator for a given target instance) at the positions of the target class in the schedule. We note inverse functions for the position effect on the potential knowledge acquisition and human error, depending upon the manipulation. The high micro-average error rate at the third position for slip error type supports the distinction with respect to the placement of the target instance. We also ran a paired two-tailed *t*-test to observe any statistically significant difference between the errors of the specific positions. For the slip errors-based annotation schedule (H1), we observe $t(43) = -2.20$ with p-value of $0.03 < 0.05$, whereas we observe $t(37) = 1.87$ with p-value of $0.07 > 0.05$ for the mistake errors-based annotation schedule (H2).

6.7. Discussion

From the paired two-tailed *t*-test results mentioned above, we found a significant difference (p-value $= 0.03$) for the error between the last position and the average of the earlier two positions in a sequence that depicts slips due to potentially memory decay of the knowledge of the target class. This shows that a significant gap with no occurrences of a class indeed increases annotation errors for that class and suggests frequent reminders of the concept/class are needed in a sequence for annotation tasks, and hence, the hypothesis H1 made for slips error was accepted. Whereas, with due to the low degree of freedom (37) and p-value of $0.07 > 0.05$, we cannot accept or reject the hypothesis H2 as we cannot rule out a difference between the error at the last position and the average of the earlier two positions in the case of some annotations (p-value of 0.07). However, it is a major challenge to model the acquisition of knowledge for a concept due to a lack of information on the prior knowledge or experience level of the annotators, and hence, we could not validate the hypothesis made for the error type *mistakes*. We discuss these limitations in Section 8. Motivated by these promising results, we next describe large-scale simulations and algorithmic solutions to mitigate such human errors in the HITL-ML paradigm-based stream processing applications focusing on slips due to potential memory decay.

Table 2
Annotation errors in the three positions of the target class instance (c.f. description in Section 6), observed after 44 and 38 filtered crowdsourced responses on two types of annotation schedule respectively. The p-value indicates the statistical significance for the difference between the error at the third studied position in comparison to the union of the errors at first and second positions.

Error type	Potential cause	1st position error	2nd position error	3rd position error	p-value
Slip	decayed knowledge	0.32	0.18	0.48	0.03
Mistake	unacquired knowledge	0.13	0.25	0.18	0.07

7. Simulation-based error testing and mitigation

We simulate the annotation task in an active learning paradigm for online stream processing (Žliobaitė et al., 2014). We design a novel method for generating a dynamic annotation schedule (instance sampling and ordering) for an annotator (simulated "oracle") such that the schedule attempts to minimize human errors (Serial Ordering-induced Slips) and maximize the overall performance of the active learning paradigm.

7.1. Mitigation algorithms

Our method first samples a batch of m instances from a time interval $[t_i, t_{i+1})$ by using a conventional uncertainty sampling algorithm for an active learning paradigm, followed by applying constraints to select only n $(n < m)$ instances for annotation that minimize the potential human memory decaying error, and then, update the machine learning model for predictions in the next time interval $[t_{i+1}, t_{i+2})$. For annotations by "oracle" in the simulation, we use ground truth labels (c.f. Section 4) along with the memory decay to simulate human errors (explained later in Section 7.3). We propose three types of algorithms (the first two being the baselines) based on diverse sampling strategies for selecting instances to annotate at the end of time interval $[t_i, t_{i+1})$:

7.1.1. (Baseline) Algorithm 1: random sampling

We randomly sample n instances from the batch of m streamed instances in the recent interval of $[t_i, t_{i+1})$. We hypothesize that random sampling can address the issue of data distribution changes for concept drift by selecting an instance from any region in the concept space, although it may be inefficient to improve learning performance over time. For consistency, we use an equal number of samples for this algorithm to the number of instances sampled by the popular active learning paradigm of uncertainty sampling, as described next.

7.1.2. (Baseline) Algorithm 2: uncertainty sampling

We predict the classes of the incoming batch of instances with the current active learning algorithmic model. At the start of the time interval of $[t_i, t_{i+1})$, along with the new incoming instances, we also receive a model, which was trained with all the annotated instances before t_i. We use this model for prediction. After prediction, we select the classified instances with uncertainty in the prediction confidence – probability in the range of $[30\%, 70\%]$. We provide the uncertainty region instances to the oracle and obtain its annotations. We hypothesize that the model will become more robust if it starts learning from the cases on the decision boundary region (Winston and Brown, 1984).

7.1.3. (Proposed) Algorithm 3: error-avoidance sampling

This algorithm relies on uncertainty sampling to first select candidate instances from uncertain regions. It then discards the instances whose predicted class (from the model received at time t_i) could either add noise to the new model or tend to be forgotten by the oracle (i.e., memory decay in human learning behavior toward that class). The algorithmic flow is formally described in Fig. 4.

Specifically, at the beginning of each interval $[t_i, t_{i+1})$, we receive four information components described below:

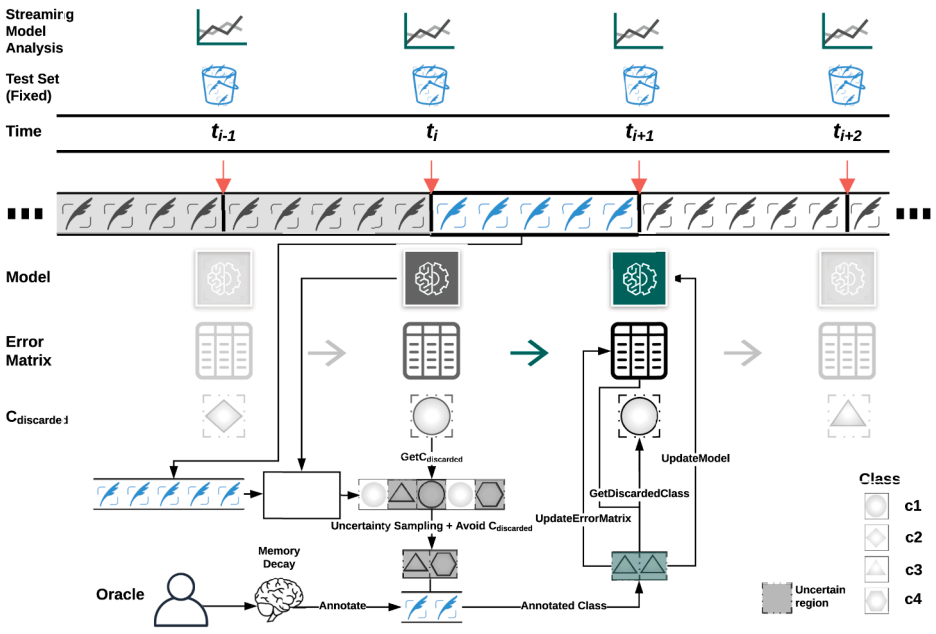

Fig. 4. Summary of the proposed Error-Avoidance Sampling-based human error mitigation algorithm. At every interval, the streaming model predicts the potential class labels with uncertainty. The uncertain instances and instances of the discarded class label are filtered out, and the remaining are annotated by the oracle, which mimics the memory decay of human annotators. Based on the new annotations, the streaming model is updated, and the next class concept to discard is computed to avoid human errors in the annotation.

- **Incoming Instances.** All incoming streams of instances at time interval t_i shown by a string of tweet symbols in Fig. 4.
- **Model.** Streaming active learning model, which is trained with all the instances that are annotated by the oracle before t_i.
- **Error Matrix.** This is a matrix that contains information about each instance annotated by the oracle from the previous two intervals (i.e., $[t_{i-2}, t_{i-1})$ and $[t_{i-1}, t_i)$) and the current interval (i.e., $[t_i, t_{i+1})$). Each row in the matrix represents an annotated instance and contains information about its arrival time, annotated class by the oracle, and the set of per-class prediction error. The per-class prediction error for a class is the average error of predictions for the annotated instances of the class present in the current error matrix, and it is computed using the active learning model updated with the current instance. For example, if an instance X has been annotated with class $c3$ by the oracle at time t_X, we store the values X, t_X, and $c3$ to the error matrix. In addition, we also have column information for per-class prediction error as $E(c_i|c_j)$ where $i \epsilon [1, num\_class]$, $j \epsilon [1, num\_class]$, and $i \neq j$. In our case $num\_class = 4$. In this example, all values of $E(c_i|c_j)$ for $i \epsilon [1, 4]$ and $j \epsilon \{1, 2, 4\}$ are copied from the previous row to the current row for time t_X, as the current instance that has arrived is annotated with class $c3$ (i.e., $j = 3$). Next we calculate $E(c_i|c3)$ for $i \epsilon [1, 4]$.

  For computing $E(c_i|c3)$ at time t_X, we predict all previous instances in the error matrix with the updated active learning model (retrained with the annotated instance X). Next, we calculate the per-class F-measure ($F\_measure(c_i)$) where the annotated class is considered as true class to compare with the predicted class, and thus, we compute per-class prediction error ($1 - F\_measure(c_i)$). Finally, we get the per-class prediction error $E(c_i|c3)$ for each class c_i in the final row of the error matrix for the instance X. This error matrix helps in determining a class to discard as explained next.
- **Discarded Class ($C_{discarded}$).** This represents the class that has induced the most errors to the other classes or whose instances appear very frequently, causing memory decay for instances of other classes. To identify the discarded class $C_{discarded}$, we first compute the *error avoidance score* and *decay score* for each class to get its final score (explained below) and then, choose the class which has the highest final score.

Algorithm steps. We now go through the flow of our proposed algorithm summarized in Fig. 4 and refer to corresponding functions and variables in parenthesis. First, for each instance X (represented by a tweet symbol in Fig. 4), we predict its class based on our current model received at t_i. Second, we select the instances that are in the uncertain region (dark-colored tweet symbols) and are not predicted with the class that is $C_{discarded}$. We believe that at each interval, $C_{discarded}$ represents a class whose instances cause error in the active learning model. Third, we schedule the selected instances for annotation by the oracle and update our model (*UpdateModel* function). Finally, we update the error matrix by adding a row for instance X, and storing values as per the error matrix definition.

To decide which class to discard, we compute two scores: *error avoidance score* and *decay score*. The error avoidance score determines the total error induced in the model for other classes due to the addition of the current instance into its training set. While the decay score determines the class that appears with excessive frequency in the stream, causing memory decay for other classes and thus, leads to annotation error. Note that we use the error matrix to decide the classes to discard only after the first three intervals.

We calculate the error avoidance score for each class c_j ($j \epsilon [1, 4]$) as:

$$GetErrorAvoidanceScore(c_j) = \sum_{k=0}^{m} \sum_{i=0}^{n} E_{k,(c_i,c_j)} \tag{3}$$

where k is the total number of instances in the error matrix.

Next, we calculate the memory decay score to determine which class appears too frequently in the stream. For each class c_j, we calculate the score as:

$$GetDecayScore(c_j) = e^{-\Delta T_j} \tag{4}$$

where ΔT_j is the time difference from the recent two occurrences of the instances of class c_j in the error matrix. Note that the *GetDecayScore* function is a simplified version of the decaying score function defined in Eq. (1). We have not parametrized this function as we generalize this decaying factor the same for all humans. Hence the memory decay score only depends on the time since the last viewed class instance irrespective of the different decaying intensities of different humans. Additionally, we observe that the memory decay score for any class is reset every time that class instance gets picked for annotation. Moreover, as the time difference increases, the decay score decreases exponentially similar to the behavior of memory decay in psychology, which has been discussed in the lab-scale experiment in Section 5.

Lastly, the final score for each class c_j is defined as:

$$Score_{c_j} = GetErrorAvoidanceScore(c_j) \times GetDecayScore(c_j) \quad (5)$$

Once we calculate the final score for each class, we determine the class c_j with the highest score as the error-inducing class to discard (*GetDiscardedClass* function). We observe that while choosing the discarded class at each interval, we are not only looking for the memory decay factor but also the input text of the instance and the effect it has on the model performance for other class labels.

7.2. Simulation experiments

We describe the data preparation for the simulated stream processing task and the active learning paradigm.

Data preparation We use labeled datasets from three major hurricanes in Central and North America as described in Section 4. We split the data into training, test, and warm-up sets. Twenty percent of the whole dataset is used as a test set. From the remaining 80% of the data, we randomly picked n instances ($n = 20$) of each class to create a warm-up set; the rest constitutes the training set. As we have a class imbalance in our data, we use an equal number of instances across classes for creating our warm-up phase model for robustness. The training data are sorted based on the arrival time of an instance (tweet) in the stream. After sorting, we divided the data into equal bins of size N. At each interval, N instances would arrive for annotation and get filtered for inclusion in the training set based on our mitigation algorithms.

We fix N based on volume as our labeled dataset is not continuous in a real-time setting but is distributed over an extended period, given it was annotated through a crowdsourcing method in prior work. Hence, we cannot fix N based on time units (seconds, minutes, etc.), but our approach is generic and applicable for other scenarios.

7.3. Active learning environment

We implement the active learning paradigm following previous work (Žliobaitė et al., 2014). First, we train the base model with the warm-up set and then, keep updating with the new incoming instances sampled by our baseline or proposed algorithms.

We used a fairly standard set-up for text classification, using pre-trained GloVe-Twitter embeddings (Pennington et al., 2014) with 200 dimensions for generating word-level features and then averaging the word-level embeddings to represent tweet-level features. We train a linear SVM model and measure the performance on the fixed test set.

For every interval t_i, we receive N instances for seeking the annotator feedback to acquire more labeled data for retraining the current model. Depending upon the mitigation algorithms, i.e. Random, Uncertainty, or Error-Avoidance Sampling; we sample the instances to obtain annotations from the oracle. To mimic human behavior, the label for the instance given by the oracle annotator is not always correct. Based on the lab-scale experiment's discussion in Section 5, we note that the error probability score for annotators making errors in annotation can be depicted using a parameterized sigmoid function in our annotation task. Thus, we utilize the value of a sigmoid function with different parameters to find the probability that the oracle generates a correct or erroneous label due to memory decay of the class as given in the formulation of Eq. (2). We define the "Memory Decay" component in Fig. 4 precisely to the oracle to highlight this memory decay behavior of the oracle. We use 3 different parameter settings to add errors through the memory decay behavior of the oracle (annotator):

1) **Slow decaying:** computes a sigmoid function with parameters estimated from errors observed in the crowd experiment: $\alpha = 0.0434$, $\lambda = 0.9025$, and $\gamma = 0.75$.
2) **Fast decaying:** uses a sigmoid function that converges to 1 faster than the slow decaying and induces errors more frequently: $\alpha = 0.03$, $\lambda = 1.00$, and $\gamma = 1.00$.
3) **No decaying:** assumes that our oracle always gives the correct labels and does not have any memory decay of the knowledge of any class. Hence, we use the true ground truth labels for each annotation.

7.4. Results

We experimented across three event datasets for a robust evaluation of our simulation algorithms. These event datasets have a varying number of instances per interval: Hurricane Harvey has $N = 36$, Irma has $N = 59$, and Maria has $N = 18$. Therefore, our results have taken into account the different burstiness of the streaming data instances during the real disaster event. We report the AUC scores for every experiment on the fixed test set per event. Fig. 5 shows the AUC scores of our three mitigation algorithms using different decay behavior settings of the oracle annotator, across all three datasets. We computed the micro average of the AUC scores at each time interval for different mitigation settings as the model is trained differently on each of them. The behaviors of accuracy and F-measure follow a similar pattern to those of AUC, thus, we omit those figures for brevity; however, they are included in the supplementary materials for transparency. The results demonstrate the effectiveness of our proposed annotation scheduling approach in contrast to the two baselines for mitigating annotation errors, and thus, improve the automatic classification performance.

7.5. Discussion

We mimicked a real-world annotation scenario by inducing different types of memory decay-based human errors (slow vs. fast decaying) in a simulated annotation schedule. The error mitigation algorithm based on our error-avoidance sampling technique can select instances for a human to annotate, which mitigates the effect of human memory decay and improves AUC scores over time across all the event datasets, despite varying numbers of instances per interval. Also, for the first three intervals, both the simple uncertainty-based and error-avoidance sampling-based algorithms are equivalent in performance during the initial time. This is consistent with an interpretation in which our algorithm may still be learning about the class that induces errors to other classes or make other classes forgotten by the annotator. Both of these algorithms show a gradual increment of performance as the new instances arrive, compared to the random sampling algorithm with highly variant behavior of the learning model. These observations support the claim that our proposed algorithm could help improve active learning paradigm-based real-time systems.

In the case of a no-decaying simulation setting, where the oracle always (but unrealistically) provides the correct label, all mitigation algorithms perform similarly to each other. This is possible due to similarity in frequency and the amount of correct oracle feedback, which constantly updates the model with new training data that gradually improves on the test set.

In the case of our error-avoidance sampling-based algorithm, the chances of inducing human error decrease due to accounting for the likelihood of memory decay of the knowledge about a class, which attempts to reduce the expected errors in annotating instances of all classes. Through the *GetDecayScore* function in Eq. (4), we give more weight to the class to discard whose instances are appearing more often in the streaming data. Hence, those class instances will be discarded in the next period so that the annotators do not forget the other classes that are appearing less frequently. Moreover, as the *GetDecayScore* function is independent in our proposed algorithm, it can take any memory decay function making our error-avoidance sampling-based algorithm generic.

In summary, our study of human error types in the annotation of streaming data presents novel insights on their effect on the performance of active learning systems in the (HITL-ML) paradigm-based

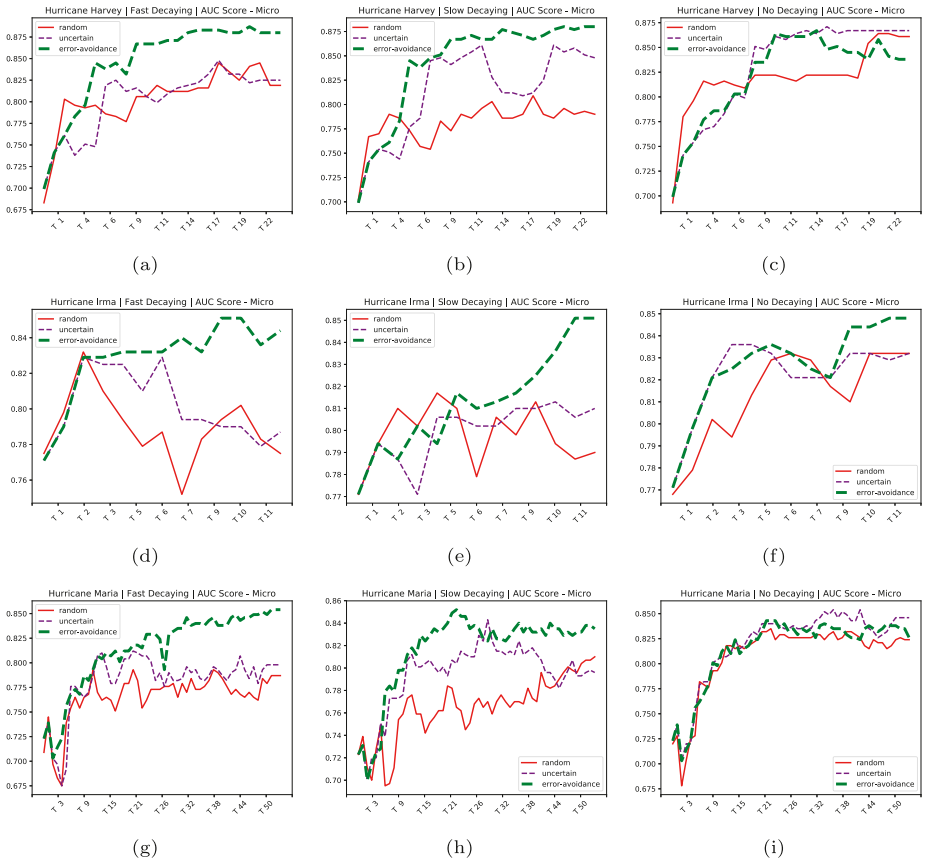

Fig. 5. AUC score of mitigation algorithms for three hurricane datasets by various decay settings, showing superior performance of the proposed error-avoidance sampling in the case of memory decay errors.

stream processing systems. This study raises awareness of instance ordering when designing crowdsourcing-based tasks. In particular, we recommend using an annotation schedule in an active learning paradigm-based system that can help reduce human errors. To the best of our knowledge, this is the first study to investigate a principled framework for quantifying human annotation errors for social stream processing and develop mitigation methods by better understanding the human annotation task as a psychological process.

8. Conclusions

We defined a framework of human errors, including mistakes and slips in the context of stream processing, based on psychological theories of human errors. We specifically focused on a quantitative model of memory decay behavior in the context of annotation tasks of humans, given that it is a common cause for both mistakes and slips. We validated the existence of memory decay-based annotation errors in a variety of experimental setups from lab-scale to crowdsourcing and provided evidence for the conceptual distinction between slips and mistakes for stream processing applications. We performed simulation-based studies to test a novel error mitigation algorithm targeted to slips that minimizes the likelihood of memory decay in an active learning paradigm-based human annotation task. The proposed method for human error mitigation can help design Human-AI collaboration systems for efficient stream processing for social media and web data in general. Such systems would require not only fewer human annotations but also reduce errors and decrease annotator memory decay.

Limitations and future work. We have provided a proof-of-concept using an over-simplified model of human memory (Anderson, 2000). In particular, we have simplified the activation and decay functions and the self-reinforcing effect of classification on persisting knowledge of class concepts in an annotation task. Different error probability score functions other than sigmoid may better model the memory decay behavior of humans. Our approach to the characterization of human annotation error is also focused on cognition and ignorant of exogenous influences on cognition, including the physical and social environment (Hollnagel, 1998). We do not claim that this study covers all types of human annotation errors in stream processing, in particular, the knowledge modification problem posed by changes in streaming content. In focusing on serial effects, we have ignored the effect of absolute time.

Our small crowd-scale study provided insufficient power to definitively distinguish the mistake-based error. A future large-scale crowdsourced study could provide more definitive support for the effectiveness of our proposed error mitigation algorithm. Nevertheless, we have documented that blind confidence in human annotation as a gold standard is gravely erroneous. Dramatic performance improvement results when the annotation is utilized with an appreciation for the human processes that generated it and might lead to errors. We hope that our framework provides a foundation for studying diverse types of annotation errors and causes, beyond text to image object recognition for a variety of stream processing applications, such as addressing burnout or inattentive worker errors in the future for human-AI teaming.

Reproducibility. Human annotations and code implementations are available upon request for research purposes.

CRediT authorship contribution statement

Rahul Pandey: Conceptualization, Methodology, Data curation, Writing – original draft, Visualization. **Hemant Purohit:** Conceptualization, Methodology, Formal analysis, Investigation, Writing – original draft, Funding acquisition. **Carlos Castillo:** Conceptualization, Formal

analysis, Validation, Writing – review & editing, Supervision. **Valerie L. Shalin:** Investigation, Resources, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Purohit thanks U.S. National Science Foundation grant awards IIS-1657379 and 1815459, and Castillo thanks the funding received from the "la Caixa" Foundation (ID 100010434), under the agreement LCF/PR/PR16/51110009 by the HUMAINT program (Human Behavior and Machine Intelligence), Joint Research Centre, European Commission for partial support to this research. The opinions expressed are those of the authors and do not reflect those of the sponsors.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.ijhcs.2022.102772

References

Alam, F., Ofli, F., Imran, M., 2018. CrisisMMD: multimodal Twitter datasets from natural disasters. Twelfth International AAAI Conference on Web and Social Media.https://www.aaai.org/ocs/index.php/ICWSM/ICWSM18/paper/view/17816

Almeida, P.R.L., Oliveira, L.S., Britto, A.S., Sabourin, R., 2018. Adapting dynamic classifier selection for concept drift. Expert Syst. Appl. 104, 67–85. https://doi.org/10.1016/j.eswa.2018.03.021.http://www.sciencedirect.com/science/article/pii/S0957417418301611

Anderson, J.R., 2000. Learning and Memory: An integrated Approach. John Wiley & Sons Inc.

Anderson, J.R., Schooler, L.J., 1991. Reflections of the environment in memory. Psychol. Sci. 2 (6), 396–408.

Bifet, A., Gavaldá, R., 2006. Kalman filters and adaptive windows for learning in data streams. In: Todorovski, L., Lavrač, N., Jantke, K.P. (Eds.), Discovery Science. Springer, Berlin, Heidelberg, pp. 29–40. https://doi.org/10.1007/11893318_7.

Brown, J., 1958. Some tests of the decay theory of immediate memory. Q. J. Exp. Psychol. 10 (1), 12–21. https://doi.org/10.1080/17470215808416249.Publisher: SAGE Publications

Bröder, A., Malejka, S., 2017. On a problematic procedure to manipulate response biases in recognition experiments: the case of implied base rates. Memory 25 (6), 736–743. https://doi.org/10.1080/09658211.2016.1214735.Publisher: Routledge _eprint:

Burghardt, K., Hogg, T., Lerman, K., 2018. Quantifying the impact of cognitive biases in question-answering systems. Twelfth International AAAI Conference on Web and Social Media.https://www.aaai.org/ocs/index.php/ICWSM/ICWSM18/paper/view/17797

Castillo, C., 2016. Big Crisis Data: Social Media in Disasters and Time-Critical Situations. Cambridge University Press.Google-Books-ID: qNqgDAAAQBAJ

Cheng, J., Cosley, D., 2013. How annotation styles influence content and preferences. Proceedings of the 24th ACM Conference on Hypertext and Social Media. Association for Computing Machinery, Paris, France, pp. 214–218. https://doi.org/10.1145/2481492.2481519.

Creswell, J.W., Poth, C.N., 2016. Qualitative Inquiry and Research Design: Choosing Among Five Approaches. SAGE Publications.Google-Books-ID: DLbBDQAAQBAJ

Delany, S.J., Cunningham, P., Tsymbal, A., Coyle, L., 2005. A case-based technique for tracking concept drift in spam filtering. In: Macintosh, A., Ellis, R., Allen, T. (Eds.), Applications and Innovations in Intelligent Systems XII. Springer, London, pp. 3–16. https://doi.org/10.1007/1-84628-103-2_1.

Ebbinghaus, H., 1913. Memory: A Contribution to Experimental Psychology, vol. 3. Teachers College, Columbia University.

Gama, J., Fernandes, R., Rocha, R., 2006. Decision trees for mining data streams. Intell. Data Anal. 10 (1), 23–45. https://doi.org/10.3233/IDA-2006-10103.Publisher: IOS Press

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A., 2014. A survey on concept drift adaptation. ACM Comput. Surv. 46 (4), 44:1–44:37. https://doi.org/10.1145/2523813.

Grant, M.J., Button, C.M., Snook, B., 2017. An evaluation of interrater reliability measures on binary tasks using *d*-prime. Appl. Psychol. Meas. 41 (4), 264–276. https://doi.org/10.1177/0146621616684584.Publisher: SAGE Publications Inc

Hansen, D.L., Schone, P.J., Corey, D., Reid, M., Gehring, J., 2013. Quality control mechanisms for crowdsourcing: peer review, arbitration, & expertise at familysearch indexing. Proceedings of the 2013 Conference on Computer Supported Cooperative Work. Association for Computing Machinery, San Antonio, Texas, USA, pp. 649–660. https://doi.org/10.1145/2441776.2441848.

Helmbold, D.P., Long, P.M., 1994. Tracking drifting concepts by minimizing disagreements. Mach. Learn. 14 (1), 27–45. https://doi.org/10.1007/BF00993161.

Hollnagel, E., 1998. Cognitive Reliability and Error Analysis Method (CREAM). Elsevier.

Hulten, G., Spencer, L., Domingos, P., 2001. Mining time-changing data streams. Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, San Francisco, California, pp. 97–106. https://doi.org/10.1145/502512.502529.

Ikonomovska, E., Gama, J., Džeroski, S., 2011. Learning model trees from evolving data streams. Data Min. Knowl. Discov. 23 (1), 128–168. https://doi.org/10.1007/s10618-010-0201-y.

Imran, M., Lykourentzou, I., Naudet, Y., Castillo, C., 2013. Engineering crowdsourced stream processing systems. arXiv preprint arXiv:1310.5463.

Jacoby, L.L., Hessels, S., Bopp, K., 2001. Proactive and retroactive effects in memory performance: dissociating recollection and accessibility bias. The Nature of Remembering: Essays in Honor of Robert G. Crowder. American Psychological Association, Washington, DC, US, pp. 35–54. https://doi.org/10.1037/10394-003.

Kahneman, D., Tversky, A., 1979. Prospect theory: an analysis of decision under risk. Econometrica 47 (2), 263–291. https://doi.org/10.2307/1914185.Publisher: [Wiley, Econometric Society]. https://www.jstor.org/stable/1914185

Klinkenberg, R., 2004. Learning drifting concepts: example selection vs. example weighting. Intell. Data Anal. 8 (3), 281–300.

Klinkenberg, R., Joachims, T., 2000. Detecting concept drift with support vector machines. ICML, pp. 487–494.

Klinkenberg, R., Renz, I., 1998. Adaptive information filtering: learning in the presence of concept drifts. Learning for text Categorization, pp. 33–40.

Koren, Y., 2010. Collaborative filtering with temporal dynamics. Commun. ACM 53 (4), 89–97. https://doi.org/10.1145/1721654.1721677.

Koychev, I., 2000. Gradual Forgetting for Adaptation to Concept DriftURL: https://research.uni-sofia.bg/handle/10506/57. Accepted: 2008-01-18T15:13:52Z Publisher: Proceedings of ECAI 2000 Workshop on Current Issues in Spatio-Temporal Reasoning.

Koychev, I., 2002. Tracking changing user interests through prior-learning of context. In: De Bra, P., Brusilovsky, P., Conejo, R. (Eds.), Adaptive Hypermedia and Adaptive Web-Based Systems. Springer, Berlin, Heidelberg, pp. 223–232. https://doi.org/10.1007/3-540-47952-X_24.

Lanquillon, C., 2001. Enhancing text classification to improve information filtering.

Lofi, C., Maarry, K.E., 2014. Design patterns for hybrid algorithmic-crowdsourcing workflows. 2014 IEEE 16th Conference on Business Informatics, vol. 1, pp. 1–8. https://doi.org/10.1109/CBI.2014.16.ISSN: 2378-1971

Loftus, G.R., 1985. Evaluating forgetting curves. J. Exp. Psychol. 11 (2), 397.

Mackworth, N.H., 1948. The breakdown of vigilance during prolonged visual search. Q. J. Exp. Psychol. 1 (1), 6–21. https://doi.org/10.1080/17470214808416738. Publisher: Routledge eprint:

Marshall, C.C., Shipman, F.M., 2013. Experiences surveying the crowd: reflections on methods, participation, and reliability. Proceedings of the 5th Annual ACM Web Science Conference. Association for Computing Machinery, Paris, France, pp. 234–243. https://doi.org/10.1145/2464464.2464485.

Melton, A.W., 1963. Implications of short-term memory for a general theory of memory. J. Verbal Learn. Verbal Behav. 2 (1), 1–21. https://doi.org/10.1016/S0022-5371(63)80063-8.https://www.sciencedirect.com/science/article/pii/S0022537163800638

Murdock Jr., B.B., 1962. The serial position effect of free recall. J. Exp. Psychol. 64 (5), 482.

Ng, W., Dash, M., 2008. A test paradigm for detecting changes in transactional data streams. In: Haritsa, J.R., Kotagiri, R., Pudi, V. (Eds.), Database Systems for Advanced Applications. Springer, Berlin, Heidelberg, pp. 204–219. https://doi.org/10.1007/978-3-540-78568-2_17.

Nguyen, D., Trieschnigg, D., Doğruöz, A.S., Gravel, R., Theune, M., Meder, T., de Jong, F., 2014. Why gender and age prediction from tweets is hard: lessons from a crowdsourcing experiment. Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. Dublin City University and Association for Computational Linguistics, Dublin, Ireland, pp. 1950–1961. https://www.aclweb.org/anthology/C14-1184.https://www.aclweb.org/anthology/C14-1184

Norman, D.A., 1981. Categorization of action slips. Psychol. Rev. 88 (1), 1.

Olteanu, A., Vieweg, S., Castillo, C., 2015. What to expect when the unexpected happens: social media communications across crises. Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing. Association for Computing Machinery, Vancouver, BC, Canada, pp. 994–1009. https://doi.org/10.1145/2675133.2675242.

Pandey, R., Castillo, C., Purohit, H., 2019. Modeling human annotation errors to design bias-aware systems for social stream processing. Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. Association for Computing Machinery, Vancouver, British Columbia, Canada, pp. 374–377. https://doi.org/10.1145/3341161.3342931.

Pennington, J., Socher, R., Manning, C., 2014. Glove: global vectors for word representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pp. 1532–1543. https://doi.org/10.3115/v1/D14-1162.https://www.aclweb.org/anthology/D14-1162

Peterson, L., Peterson, M.J., 1959. Short-term retention of individual verbal items. J. Exp. Psychol. 58 (3), 193–198. https://doi.org/10.1037/h0049234.Place: US Publisher: American Psychological Association

Purohit, H., Castillo, C., Imran, M., Pandey, R., 2018. Ranking of social media alerts with workload bounds in emergency operation centers. 2018 IEEE/WIC/ACM

International Conference on Web Intelligence (WI), pp. 206–213. https://doi.org/10.1109/WI.2018.00-88.

Purohit, H., Castillo, C., Imran, M., Pandey, R., 2018. Social-EOC: serviceability model to rank social media requests for emergency operation centers. 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 119–126. https://doi.org/10.1109/ASONAM.2018.8508709.ISSN: 2473-991X

Reason, J., 2000. Human error: models and management. BMJ 320 (7237), 768–770. https://doi.org/10.1136/bmj.320.7237.768.Publisher: British Medical Journal Publishing Group Section: Education and debate. URL: https://www.bmj.com/content/320/7237/768

Salganicoff, M., 1993. Density-adaptive learning and forgetting. Proceedings of the Tenth International Conference on International Conference on Machine Learning. Morgan Kaufmann Publishers Inc., Amherst, MA, USA, pp. 276–283.

Shirky, C., 2008. It's not information overload. It's filter failure. Web. September.

Sutton, J., League, C., Sellnow, T.L., Sellnow, D.D., 2015. Terse messaging and public health in the midst of natural disasters: the case of the boulder floods. Health Commun. 30 (2), 135–143. https://doi.org/10.1080/10410236.2014.974124. Publisher: Routledge _eprint:

Tulving, E., Pearlstone, Z., 1966. Availability versus accessibility of information in memory for words. J. Verbal Learn. Verbal Behav. 5 (4), 381–391. https://doi.org/10.1016/S0022-5371(66)80048-8.http://www.sciencedirect.com/science/article/pii/S0022537166800488

Vitter, J.S., 1985. Random sampling with a reservoir. ACM Trans. Math. Softw. 11 (1), 37–57. https://doi.org/10.1145/3147.3165.

Widmer, G., Kubat, M., 1996. Learning in the presence of concept drift and hidden contexts. Mach. Learn. 23 (1), 69–101. https://doi.org/10.1023/A:1018046501280.

Wiener, E.L., 1987. Application of vigilance research: rare, medium, or well done? Hum. Factors 29 (6), 725–736. https://doi.org/10.1177/001872088702900611.Publisher: SAGE Publications Inc

Winston, P.H., Brown, R.H., 1984. Artificial Intelligence: An MIT Perspective. MIT Press.

Yao, R., Shi, Q., Shen, C., Zhang, Y., van den Hengel, A., 2012. Robust tracking with weighted online structured learning. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (Eds.), Computer Vision - ECCV 2012. Springer, Berlin, Heidelberg, pp. 158–172. https://doi.org/10.1007/978-3-642-33712-3_12.

Zhang, J., Patel, V.L., Johnson, T.R., Shortliffe, E.H., 2004. A cognitive taxonomy of medical errors. J. Biomed. Inform. 37 (3), 193–204. https://doi.org/10.1016/j.jbi.2004.04.004.http://www.sciencedirect.com/science/article/pii/S1532046404000528

Zhao, P., Hoi, S.C.H., Jin, R., Yang, T., 2011. Online AUC maximization. Proceedings of the 28th International Conference on Machine Learning ICML 2011: Bellevue, WA, June 28–July 2, pp. 233–240.https://ink.library.smu.edu.sg/sis_research/2351

Zhou, C., Xiu, H., Wang, Y., Yu, X., 2021. Characterizing the dissemination of misinformation on social media in health emergencies: an empirical study based on COVID-19. Inf. Process. Manag. 58 (4), 102554. https://doi.org/10.1016/j.ipm.2021.102554.https://www.sciencedirect.com/science/article/pii/S0306457321000583

Žliobaitė, I., 2011. Combining similarity in time and space for training set formation under concept drift. Intell. Data Anal. 15 (4), 589–611. https://doi.org/10.3233/IDA-2011-0484.Publisher: IOS Press

Žliobaitė, I., Bifet, A., Pfahringer, B., Holmes, G., 2014. Active learning with drifting streaming data. IEEE Trans. Neural Netw. Learn. Syst. 25 (1), 27–39. https://doi.org/10.1109/TNNLS.2012.2236570.Conference Name: IEEE Transactions on Neural Networks and Learning Systems