Improving the Dark Web Classifier with Active Learning and Annotation Error Detection

Pablo San Gil[1,2], Romana Pernisch[1,3] Eljo Haspels[2], and Mark van Staalduinen[2]

[1] Vrije Universiteit Amsterdam, Amsterdam, the Netherlands
[2] CFLW Cyber Strategies, The Hague, the Netherlands
[3] Discovery Lab, Elsevier, Amsterdam, the Netherlands
pablo.sangilsatrustegui@cflw.com, r.pernisch@vu.nl, eljo.haspels@cflw.com, mark.vanstaalduinen@cflw.com

Abstract. Active learning (AL) methods aim to reduce the human labeling effort by selecting the most significant unlabeled samples. Annotation error detection (AED) strategies aim to identify noisy samples in the dataset. In this work, we tackle these objectives together in the context of multi-label classification of Dark Web pages, in order to label and introduce new data, and to update a Dark Web classifier. Here, only a small portion of the data contains correct labels and new pages need to be labeled and corrected ad-hoc. To do so, a Human-in-the-loop pipeline with AL and AED strategies is applied to a base model, a multi-label Dark Web content classifier. In this work, we present the first comprehensive survey of many AL and AED strategies tested on a real-world dataset. We found that Mean Max Loss performed best as the AL strategy and Datamap as the AED approach.

Keywords: active learning · annotation error detection · text classification · multi-label classification · Dark Web

1 Introduction

The Dark Web is a subsection of the internet, with encrypted communication over private or peer-to-peer networks, accessible via specialized software, favoring anonymity and privacy [37]. The largest Dark Web service, The Onion Router[4] (Tor), reported over 4.5 million connected users as of 20-08-2023 [32]. Dark Web services are used for different purposes. They can be used for normal internet activity, for people that highly esteem privacy, to avoid censorship or to share sensitive information. However, it can also provide cover for illegal activities, letting criminals act without being tracked by law enforcement agencies [5].

CFLW Cyber Strategies[5] is a company focused on providing intelligence services to increase cyberspace safety. One of their products is the *Dark Web Monitor*, an open-source intelligence repository that provides insights into criminal

[4] https://www.torproject.org/
[5] https://cflw.com/

activities facilitated in the dark web. It contains HTML snapshots of dark web domains, which are updated regularly in case of domain modification. CFLW extracts characteristics of each domain, such as status (online or offline), mentioned email addresses, crypto assets, PGP keys, total number of pages and a set of labels that describe its content. Investigators use this information to filter out irrelevant domains to make their work more efficient. Currently, CFLW annotates all domains by hand. However, manual labelling requires human experts, is time-consuming and error prone, because of the repetitiveness of the task. CFLW already implemented an automated labelling system, a multi-label Dark Web classifier [5]. Although performing reasonably well, the amount of training data is insufficient and imbalanced, making the model suboptimal. *Active learning* (AL) techniques can help reduce the labelling cost, and in turn *Annotation Error Detection* (AED) could help to review the data and improve its quality. Hence, we address the problem of multi-label classification of Dark Web pages, where the correctness of the data labels is unknown and new pages need to be labeled and corrected ad hoc in a human-in-the-loop approach. To reach this ultimate goal, we investigate the AL and AED strategies in seperate steps.

In order to design our pipeline, we first introduce a new AL strategy called *Dual active learning based on Uncertainty and Cosine Similarity* (DALUCS) and answer the first research question:

RQ1: *To what extent is DALUCS successful in optimizing the query strategy to minimize human labeling efforts while maintaining data quality?*

As a second step, we thoroughly investigate and compare different state-of-the-art AL, including DALUCS, to find the most suitable for the presented usecase. To the best of our knowledge, we present the most extensive comparison of the different approaches in this work on a real-world dataset, hence, answering the second research question:

RQ2: *Which state-of-the-art Active Learning strategies can best optimize the query strategy to minimize human labeling efforts while maintaining data quality on the Dark Web Monitor dataset?*

We evaluate the different strategies in four experiments, that simulate a classic Active Learning setup as well as cases where there is data drift. We also answer to practical questions about the optimal usage of Active Learning, such as the sampling time or the best sampling size. We found, that even though DALUCS performs as intended, it does not outperform simpler state-of-the-art AL strategies. Hence, we select Mean Max Loss strategy, that provided near-optimal results at a low computational cost.

To finish building our pipeline, we also need to investigate AED approaches. First, we introduce our own approach called *Distance-based Label Error Detection* (DLED). Similarly to DALUCS, we answer the following research question:

RQ3: *To what extent can DLED effectively mitigate the impact of noisy data by detecting mislabeled instances?*

Second, we compare various AED approaches, to select the most suitable one, by answering the last reasearch question:

RQ4: *Which Annotation Error Detection methods can most effectively mitigate the impact of noisy data by detecting mislabeled instances?*

We evaluate the different strategies in three experiments. In the first two we introduce artificial noise to the dataset and predict the noisy samples using the different AED strategies, assessing their ability to find them and the impact of correcting them on the model. In the third we estimate the real noise rate in our dataset, before and after correcting the noisy samples found by our AED strategy. In the second half of our investigation, we found that DLED did not outperform other state-of-the-art, but Datamap Confidence proved to be an efficient method to reduce the noise in the dataset. By answering our four research questions, we implement a final pipeline which integrates both AL and AED for CFLW using the best performing approaches, namely Mean Max Loss and Datamap Confidence.

To sum up, in this work we contribute the following insights

- A comprehensive comparison of Active Learning strategies on a real-world dataset under varing conditions of (1) label imbalance, (2) label shift, with Mean Max Loss being the best performing one.
- A comprehensive comparison of Annotation Error Detection approaches on a real-world dataset, with Datamap Confidence being the best performing one.
- An analysis of the implications of querying with Active Learning techniques in datasets containing noisy samples, and the importance of combining it with Error Detection.
- A report of the limitations of the application of some Error Detection approaches on dynamic datasets.

The rest of this paper is structured in the following way. We present related work on AL and AED in Section 2. Then, we introduce the methodology. In Section 4, we give details about the data and experiments, followed by the results and findings in Section 5. We then present limitations and future work, and conclude this paper in Section 6

2 Related work

AL and AED have garnered significant attention from the research community in recent years. However, limited attention has been given to the combination of these techniques. Let us review the current status of these domains and presents state-of-the-art methods.

Active Learning AL research focuses on situations in which there is a small labeled dataset available compared to the amount of unlabeled data, and where labeling is a costly practice. Therefore, it is desirable to reduce the labeling costs by only querying those samples that induce a significant improvement to

the model. Sampling strategies mainly depend on the granularity and informativeness measure used. Granularity refers to the format in which samples are selected. It can be example based, where a particular sample is selected, example-label based, where a sample and specific label of it are selected, mixed mode, where multiple samples and a subset of their labels are selected, and batch mode selection, where several samples are selected at once [34]. Informativeness measures evaluate how relevant a given sample is to the model, and determines which ones are selected [27]. They can be classified into the following categories: uncertainty measure -based on the uncertainty of the classifier on the samples-, label correlation -that measures the relevance of the labels-, representativeness -measuring the representativeness of each sample in the unlabeled dataset-, diversity -asseses the novely of a sample within the labeled dataset-, noise content -asseses the noise content of a sample- and expected model change -predicts the impact of adding a sample in the model-[34].

State-of-the-art methods combine different scoring metrics to have a more complete measure of the informativeness of the samples. Chakrabroty et al. [6] use a combination of uncertainty measure and redundancy to select the optimal batch. Li et al. [19] combine uncertainty with cardinality inconsistency (label correlation). Common algorithms such as QUIRE [12] or AUDI [13] combine uncertainty measures with representativeness and diversity methods, respectively. Reyes et al. [25] present a rank-based aggregation of uncertainty measures with label correlation for text classification purposes. BADGE [3] and ALPS [36] incorporate gradient-based uncertainty metrics along with a cluster based diversity approach. Gui et al. [1] first select instance-label pairs based on uncertainty, label correlation and label space sparsity, to later select the optimal batch based on diversity. These are only some noticeable examples from literature.

In the case of multi-label AL for transfer learning with language models such as BERT, we found two studies in the literature [7,33]. Ein-Dor et al. [7] compare uncertainty measures with expected model change and diversity methods. Meanwhile, Wertz et al. [33] investigate class embeddings with respect to state-of-the-art methods like ALPS [36] and CVIRS [25]. Nonetheless, we found a lack of systematic and complete comparison of these algorithms in the context of text classification with language models.

The motivation for this work is the use of AL for selecting new incoming data in a lifelong learning setup. Previously mentioned examples only tackle the problem in a static setup, where the initial dataset and the unlabeled dataset do not change over time or do not come from different data distributions. However, there can be a domain shift between the labeled and the unlabeled dataset. As in this work one of the objectives is to be able to update our model robustly against changes in data, it is interesting for us to look at AL methods from this perspective. In fact, Active Universal Domain Adaptation [20] is a very recent field of study. Here, we include recognized AL strategies such as CVIRS [25] or Discriminative Active Learning [8] (DAL) that take into account domain shifts in the design of the AL strategy, and more recent methods such as AUAN [20], EADA [35], AADA [29] or CLUE [22], that are specifically designed to tackle

Domain Adaptation from an AL perspective. Most of these methods rely on complex approaches or additional machine learning methods, which add time complexity. In our work, we propose a method that uses a simple and time-efficient approach to tackle AL from a domain adaptation perspective, which we introduce in Section 3.1.

Annotation Error Detection Song et al. estimated that 8 to 38% of labels are incorrect as a consequence of being manually labeled [28]. There are different approaches to reduce the impact of noise in the model. Karimi et al. [14] provided a survey on different approaches in the context of medical imaging such as: label cleaning, special network architectures, noise-robust loss functions, data re-weighting, data and label consistency, and special training procedures. In this study, human-in-the-loop based label cleaning have been studied. Thus, this approach refers to the use of the AED to find mislabeled instances within the labeled dataset. Our approach assumes that the detection of noisy samples ought to be sufficient to tackle the data noise issue.

Known as Active Label Correction, there have been several AL based AED strategies [23,4,18,15]. Klie et al. [16] provides a landscape study about existing methods for AED in Natural Language Processing (NLP), including for text classification tasks. The methods implemented and compared in our work are taken from [16], as it is, to the best of our knowledge, the most complete review of AED methods in the context of multi-label text classification. We also present as an alternative a novel distance-based flagger method presented in Section 3.2.

It is important to note that during the literature review, we did not come across any studies on AL which considered the possibility of noisy data. Nonetheless, in this work and other real world applications with noisy data, it is expected that AL methods can be keen to sample noisy data. Therefore, AED is expected to be paramount for the actual implementation of AL for lifelong learning.

3 Methodology

Figure 1 depicts the workflow of the combination of AL and AED to introduce new data and update the model. Here, a base model is initially trained on a labeled dataset. After training, a batch of unlabeled data is selected to be labeled and introduced into the model using an AL query strategy. Before adding this new data to our dataset, it is reviewed by an AED system, that detects noisy samples, in order to ensure a clean dataset. This workflow has two main objectives, first of which is data quality. Second, by iteratively selecting the most relevant data to be added, we improve the performance to the fullest extent, by selecting as little data as possible.

First, we evaluate and compare different AL strategies in three different experiments. The first experiment (*AL-normal*), consists of the typical AL setup, where the different AL strategies query new data sequentially and the model is trained with the queried data. The next experiments (*AL-slight shift* and *AL-extreme shift*), have a similar setup, but a slight or extreme data shift is introduced to test the AL strategies under more difficult conditions.

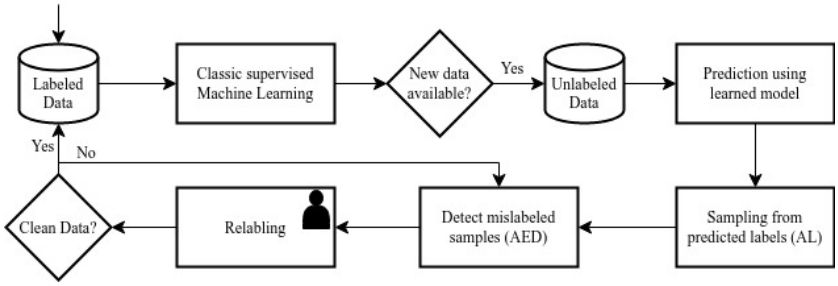


Fig. 1: Workflow of the proposed pipeline.

Second, we study AED techniques. In experiment *AED-normal*, we introduce artificial noise to the dataset, and test the ability of the different strategies to find said noise. Then, experiment *AED-estimations* consists of estimating the real error rate in the dataset by first manually examining a sample and also applying the best performing AED and analysing the returned samples in detail.

Base Model: The base model utilised during this project is the result of the study performed by Brinkhuisen [5]. The model consists of a SBERT [24] based text vectorisation, that creates context-aware embeddings from text. The embedding module is followed by a classification module. It consists of a Feed-Forward Fully Connected Neural Network with a single hidden layer that gives multi-label predictions. Further details about the model are explained in Appendix A.

3.1 Active Learning Strategies

In terms of granularity, in this project, every class is equally important for every sample, therefore, we use "example based" rather than "example-label" methods. However, as our base model is very data hungry, batch-mode approaches are of interest. Therefore, we adapted example-based algorithms to sample in batch mode. We disregard ensemble methods because they are based on uncertainty averaging over a series of models, that is, they require the training of several models, which is undesirable in our use case. Bayesian AL methods have also been discarded as they do not scale well for large datasets [26].

As baseline, we use a random sampling method. Next, we selected some classic AL methods, specifically Least Confidence, Mean/Max Entropy, Max Score and Mean Max Loss. Further, we also chose state-of-the-art techniques such as Expected Gradient Length [11] (EGL), BADGE [3], CLUE [22] and Discriminative Active Learning [8] (DAL). We implemented these approaches our selves, following the instructions from the respective paper and the provided code repositories. Last, we propose our own AL approach called *Dual Active Learning based on Uncertainty and Cosine Similarity* (DALUCS).

Being our objective to detect data drifts, DALUCS is designed to be robust for feature space changes within new data in the unlabeled dataset. It is based on the assumption that in case of data drift the labeled data distribution is not

representative of the actual data distribution, while the unlabeled data is. This way, we want to detect samples that do not match the current distribution of the labeled dataset. Uncertainty based methods and other low-cost AL algorithms are not designed to directly tackle these changes, making them less robust in this scenario. At the same time, sampling strategies specifically designed for data drift, as CLUE [22] or DAL [8], usually rely on other machine learning algorithms, which are computationally expensive. Meanwhile, DALUCS is both robust against changes in feature space and computationally cheap.

DALUCS uses an uncertainty metric to detect examples with low predictive confidence. This metric consists of a rank-based aggregation of the separation margin of the predictions over all labels. Then, it detects changes in the feature space by comparing features in the embedding space of the unlabeled samples with features of the labeled samples using cosine similarity. Last, the scores for the uncertainty and the similarity are aggregated, giving the final score of each sample. A detailed explanation about DALUCS is available in Appendix B.

3.2 Annotation Error Detection Strategies

As we are not looking to recreate the work by Klie et al. [16], we select only the best performing strategies per group to be tested on our real-world data. Time and cost efficiency plays an important role. Therefore, we disregard ensemble methods because of their high computational cost. Thus, two model-based strategies are selected, Retag [10] and Confident Learning [21]; two vector space proximity strategies are selected, Mean Distance [17], k-Nearest Neighbor Entropy [9] and three training dynamics strategies Datamap Confidence [30], Curriculum Spotter [2] and Leitner Spotter [2]. We implemented the strategies following the descriptions from the original publications and code repositories.

Depending on the type of approach used, a strategy might not be applicable in a given setup, and, to the best of our knowledge, there are no studies reviewing these limitations. First, model-based strategies need to be applied to out of sample data [16,21], this means, data on which the model has not been trained. Thus, in our setup, model-based strategies are appropriate for reviewing newly incoming data. However, the training dataset needs reviewing with cross-validation, which can be too computationally expensive. Second, training dynamics strategies are useful to review in sample data. Thus, they can be used to review the training dataset, but would be incompatible with fine-tuning to review newly incoming data efficiently. Last, vector space proximity strategies, as they are model independent, can be used in any setting, but perform worse [16].

Another important distinction are flagger vs scorer approaches [16]. Scorers give to each sample a score that represents the likeliness of being wrongly labeled. This requires defining a threshold from which that value on we consider the sample to be wrongly labeled. On the other hand, flaggers present a binary judgment whether the labels of an instance are correct or incorrect. The implications for the implementation of these methods in a real setup have not been analyzed. Selecting samples from a score implies first determining a threshold. Appendix I shows a comparison of the performance of the same method with dif-

ferent thresholds. We found the best strategy to be selecting the n^{th} percentile of the distribution, where n is the error rate in the dataset. However, in real scenarios the error rate is not known. Therefore, we estimate it manually by analysing a subset of 1000 samples in the dataset, using Label Studio[6]. These estimations showed that there is a significant amount of noisy data in the dataset, as it can be seen in Table 2.

In order to avoid the aforementioned limitations for the application on real setups, we present a novel vector space proximity AED strategy, that is also a flagger and that implies a very low computational cost. This way, we can easily detect label errors in any setting, without the need of estimating optimal thresholds, and in a low amount of time. We call this method *Distance-based Label Error Detection* (DLED).

DLED is the flagger alternative to Mean Distance [17]. It computes the mean points in the embedding space for each class. Then, it compares the distances of each sample to all the class means. If the class of the sample corresponds to the closest mean, then the sample is considered to be correct. Otherwise, it is considered noisy. A detailed explanation of DLED is available in Appendix C

4 Experimental setup

We first study the AL strategies and AED approaches separately, based on which we then designed the final pipeline. Here, we first review the data and then explain the individual experiments. We provide all our code in a git repository[7].

4.1 Dataset

The dataset used during this study is a subset of the Dark Web Monitor repository introduced in Section 1. The data was extracted and pre-processed following the work by Brinkhuisen et al. [5]. Due to the sensitive nature of the data, access is restricted[8]. First, the data was extracted. Each sample consists of the domain ID, set of assigned tags, and HTML source code. The data extraction step includes reading the HTML files of the Dark Web domains, removing the duplicates and extracting the text within them with *BeautifulSoup*[9]. We remove the duplicates by comparing the MD5 content hash [31], the HTML element tree structure of the HTML files and the raw text was extracted. Duplicates in any of these extractions were removed, keeping only the first domain. Next, preprocessing consists of removing special characters, hyperlinks, IP addresses, etc. Then, we remove excessive white space and punctuation, and transform every letter to lowercase. We apply tokenisation as the next step, a process where the text is separated into individual words. Last, labels, which were stored as text tags, were converted to one-hot vectors.

[6] `https://labelstud.io/`

[7] `https://github.com/pablosgs/Active_Learning`

[8] For scientific purposes access can be granted after a vetting process. Please contact support@cflw.com for more information.

[9] `https://pypi.org/project/beautifulsoup4/`

The final dataset contained 13'422 samples consisting of the pre-processed text and the set of labels. These labels correspond to the main abuse types present in the Dark Web Monitor: Cyber Crime, Financial Crime, Goods and Services, Sexual Abuse and Violent Crime. All samples meet the following criteria: text must be written in English, the assigned tags contain at least one label related to an abuse type, and the HTML source code must be the most recent version of the home page. Some more insights about the data can be seen in Appendix D. The data suffers from a data imbalance, having more samples of class Financial Crime, and less of Sexual Abuse and Violent Crime. A weighted Cross Entropy function was introduced to the base model in order to mitigate the impact of this data imbalance.

4.2 Active Learning

AL-normal: First, we evaluate and compare the AL strategies. In an AL pipeline, the base model is trained sequentially introducing new batches of instances, that have been sampled by the corresponding strategy. In this case, in each round 500 samples have been selected by the AL strategy. As the objective is to maximize the performance while minimizing the number of samples, AL models are usually evaluated by plotting the performance against the number of newly sampled instances. We evaluate performance with three standard metrics: accuracy, micro-F1 and macro-F1 scores.

AL-slight shift: We are also interested in observing how the different AL models react to new domains coming from different distributions. In order to evaluate the sampling strategies in this context, we perform two experiments. First, an AL setup with a slight data drift. Here, we remove instances of one class and initially train the model with 50% of the samples belonging to the remaining five classes. The unlabeled dataset consist of the other 50% and the data belonging to the removed class. We aim to observe how the AL strategies handle this new data. To do so, the models have four sampling rounds of 100 samples each. We repeat this experiment while isolating each class individually in order to provide a fair comparison between the strategies.

AL-extreme shift: This experiment presents a more extreme shift. To do so, we progressively fill the unlabeled pile with new classes. This way, the unlabeled samples starts only having all samples belonging to one label, the model has five AL rounds of 100 instances to query from this data, after which we add samples from another label to the unlabeled set. We then repeat this process on the new data distribution. The experiment continues to cover the whole dataset.

All the AL experiments were repeated with 5 different initializations to ensure the reliability of the results.

AL-practical: Lastly, we investigate the strategies for sampling efficiency in terms of run time as well as optimal query size. We use the *AL-normal* setup.

4.3 Annotation Error Detection

AED-normal: First, we introduce artificial noise at different rates (5%, 10% and 20%) to the dataset in order to assess the efficiency of the methods. For

each rate, we detect these introduced the errors with the different algorithms. Then, we measure the precision and recall for each case as well as micro-F1 improvement after correction of the detected errors. For scorer methods, we choose the threshold, which provides the best performance for each error rate.

AED-impact: One of the objectives of having a clean dataset is to mitigate the impact of noisy data on the performance of the model. Therefore, the influence of correcting the detected errors on the model was assessed. To do so, the performance of the model with the artificially introduced error is recorded. Then, we use the different approaches to detect the noisy samples, which we subsequently correct. After the cleaning process, we retrain the model and record the new performance. Last, the initial performance is compared to the new performance. We evaluate how robust is the model to label noise after being cleaned by the different AED strategies. We repeat this experiment with the same artificially introduced errors as in experiment *AED-normal*, and at the same rates (5%, 10% and 20%). Additionally, we introduce a gold-standard for this experiment, that corresponds to cleaning all noisy samples, i.e. a perfect AED.

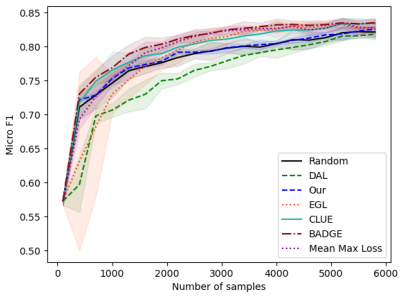
Correcting the most malicious examples implies the biggest change in the model, thus the biggest improvement in the predictions. We refer to malicious examples to the mislabeled samples where the base model is able to predict the actual correct label with a high confidence, but due to the mislabeling, it is thought to be misclassified. Therefore, they produce a high loss (which should not induce), and a big undesired change in the model's parameters.

AED-estimations: Studies in the literature usually evaluate AED strategies with experiment *AED-normal* [16]. However, real world problems do not contain any information of the error content in the data. Therefore, previous experiments can provide insights about the efficiency of the methods, but are not completely realistic. To tackle this problem, here we estimated the real error rate in the dataset, by taking a subsample of 1000 instances and manually inspecting them. Next, a simulation of the real usage of the AED algorithm was tried. To do so, we run the best performing strategy in previous experiments, Datamap Confidence [30], on our dataset, without adding any artificial errors. We then inspect the returned samples to assess the precision of the model.

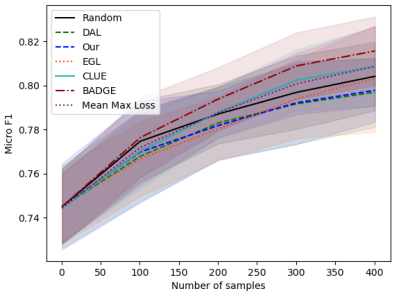
Lastly, we combine the AED with the samples selected by using AL. As AL tries to select samples based on the loss that they produce to the model, it is likely that they are keen to select samples with issues. This is a possible drawback that has not yet been covered in the literature. Therefore, it is of our interest to check if this is true and if our annotation error detection system is effective at countering this effect. Thus, the error rates of the queried samples by the AL strategy have also been estimated the same way, along with the AED of these samples. Here, we apply a multinomial 95% confidence interval estimation.

5 Results and Discussion

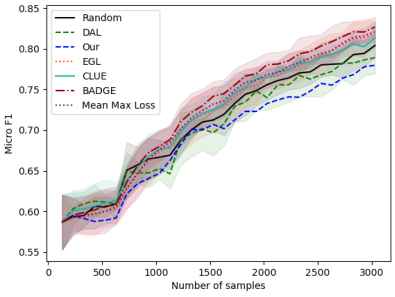
This section provides the results of the described experiments, along with a discussion and insights extracted from those. First, we show the results of the



(a) *AL-normal*



(b) *AL-slight shift*



(c) *AL-extreme shift*

Fig. 2: Performance captured with micro-F1 of the model vs number of samples. The faster the improvement of the performance, the better the quality of the data, and the query strategy. The interval depicts the variability across the different intialisations, using the standard deviation. Accuracy and macro-F1 are reported in Appendix G.

AL related experiments, followed by the results of the research on AED. Lastly, we also discuss limitations of this study and future work.

5.1 Active Learning

AL-normal: Figure 2a shows the performance of the different implemented methods in the simple AL setup. The displayed results are calculated as the mean of the results with all the different initialisations. For better readability, we only included Mean Max Loss from the uncertainty measures, as it was the best performing one. BADGE [3] is the best performing AL strategy, followed by Mean Max Loss. Additionally, all strategies including DALUCS outperform the random selection except DAL [8].

AL-slight shift: The results of the slight data drift setup can be seen in Figure 2b, which shows the performance of the AL strategies in a more challenging set-up than AL-normal. Notable is the much larger performance interval of performance for each strategy as well as the performance range for these.

Strategy	Random	MML	EGL [11]	DALUCS	DAL [8]	CLUE [22]	BADGE [3]
Time (s)	5.43 e-7	8 e-6	1.8 e-5	5.28 e-4	3.23 e-2	6.79 e-2	2.19 e-2

Table 1: Sampling time of each strategy. MML stands for Min Max Loss.

AL-extreme shift: Another class incremental setup was designed to evaluate the performance of the strategies in a more extreme case. We visualise the results in Figure 2c. As visualised, BADGE [3] is consistently the best performing AL strategy, followed by CLUE [22] and Mean Max Loss. Most strategies also outperform random sampling in this setup, including DALUCS.

However, DALUCS does not provide optimal performance. One reason for its poor performance lies in the possibility of the data distribution not being explainable using just a similarity measure. Another reason can be that trying to find unique samples may not be the best approach for this particular dataset. In fact, DAL [8] also performed poorly, which is a state-of-the-art AL strategy with also an approach based on the search for samples that are out of the current labeled data distribution.

AL-practical: We are also interested in run time and computational cost, not just performance of the strategies and we report the sampling time in Table 1. BADGE [3], CLUE [22] and DAL [8] take more time to query instances. Therefore, Mean Max Loss is considered the best practice for this context, as it provides the best balance between performance and sampling time.

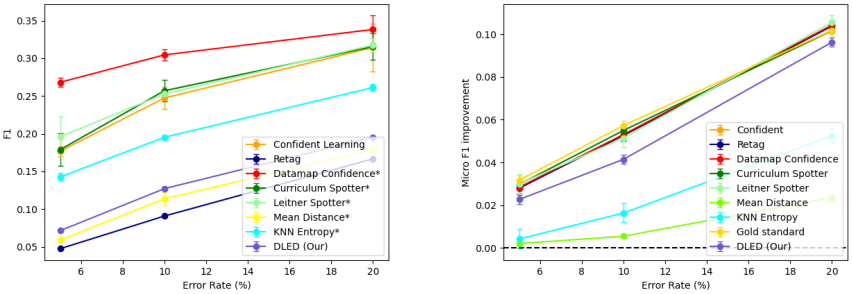
After evaluating the different sampling strategies, the optimal sampling size has been evaluated. To do so, different sizes have been tried and compared, as it can be seen in Figure 8 on Appendix F. Results suggest that the best performance is encountered with a smaller sampling size until a plateau is reached.

RQ1: Can DALUCS optimize the query strategy to minimize human labeling costs while maintaining data quality? As we discussed above, DALUCS does not provide optimal performances compared to other state-of-the-art methods. Therefore it does not optimize the query strategy.

RQ2: Which state-of-the-art Active Learning strategies can best optimize the query strategy to minimize human labeling efforts while maintaining data quality on the Dark Web Monitor dataset? Mean Max Loss provides near optimum performances, significantly better than random sampling, and a reasonable sampling time. Therefore, we can use MML to optimize the query strategy, reducing human labeling efforts. MML is not only able to maintain data quality, but even improve it. Appendix E shows that Mean Max Loss can help to fight data imbalance.

5.2 Annotation Error Detection

AED-normal: Figure 3a depicts the result of the experiment, scorer methods indicated with "*". We set the optimal threshold for scorer methods to the n^{th} percentile of the scores, with n being the percentage of correct samples. We explain our reasoning in Appendix I. It is important to note that, as explained



(a) F1-score

(b) Performance improvement after correction.

Fig. 3: Comparison of different AED methods. X-axis describes the rate of the artificially introduced errors, y-axis shows performance. (a) F1-score of the error detections by each detection method. Precision and recall are reported in Appendix H. (b) Performance improvement of the model after the correction of the detected errors by the different strategies.

in Appendix I, the need of a threshold limits the usability of these methods on real data, as the real error rate in the dataset is needed. From the model-based strategies, Confident Learning shows a remarkable performance and Retag shows a very poor precision because of its high recall. Training-dynamic based methods perform best, Datamap Confidence [30] in particular. As the impact on the model does not vary much across strategies, Datamap Confidence [30] looks to be the most robust strategy, as it provides the best precision, and a high recall. Last, in line with the literature, Vector Space Proximity strategies, including DLED, do not perform as good as strategies based on training dynamics or model-based. However, DLED outperforms other state-of-the-art Vector Space Proximity methods (Mean Distance). This shows the inefficiency of finding mislabeled samples by looking at the vector space alone, and the necessity of analysing the behavior of the model with each sample.

AED-impact: Figure 3b depicts the improvement in the performance of the model after cleaning the data with each of the methods. Except for KNN-Entropy and Mean Distance, all the strategies trigger a similar improvement in the model, even as much as the gold standard. This suggests that AED is effective at detecting the malicious samples, enabling their correction.

AED-estimations: The first row of Table 2 shows the results of the error estimation in our data based on 1000 randomly selected samples. 76% of the data is correct, and from the remaining data the great majority of errors are noisy samples. This noise corresponds to log-in pages, redirection pages, or similar ones that do not contain any information about the content. The table also shows the precision estimate of the AED with Datamap Confidence [30]. Datamap Confidence [30] performs with 71% precision, according to our estimations. From these

Type of error	Correct	Noise	Random	Difficult
Dataset	0.76 ± 0.037	0.14 ± 0.031	0.04 ± 0.016	0.06 ± 0.019
Detected	0.29 ± 0.034	0.34 ± 0.039	0.24 ± 0.029	0.13 ± 0.025
Reurned by AL	0.45 ± 0.043	0.14 ± 0.030	0.016 ± 0.032	0.09 ± 0.023
Detected	0.25 ± 0.028	0.32 ± 0.040	0.24 ± 0.037	0.19 ± 0.023

Table 2: Estimation of error rates in the data with a 95% CI. Each column represents a type of error. Correct: Sample is correctly labeled. Noise: Sample contains noise in the text, not in the label. The model should not be able to predict correctly the label. Random: Labels are clearly incorrect (random error). Difficult: The correctness of the labels is difficult to assess, even for a person.

detected samples, almost 47% correspond to noisy samples, while actual mislabeled samples correspond to only 33% of all the returned instances. However, noisy and difficult samples are also worth to be reviewed by a human annotator for correction or removal from the dataset. The third row of Table 2 shows an estimation of the error content of the samples selected by Min Max Loss and the fourth row depicts the precision estimation of the samples detected by Datamap Confidence [30]. It can be seen that more than half of the samples selected have issues, and that 75% of the flagged samples have issues.

RQ3: Can DLED effectively mitigate the impact of noisy data by detecting mislabeled instances? The estimations showed noise in the dataset and highlight the necessity of incorporating an AED system. However, DLED was not efficient at detecting noisy samples, as it was outperformed by other state-of-the-art methods. Nonetheless, being DLED the flagger alternative to Mean Distance, not only did it performed better, but also provides an easier usability than it.

RQ4: Which Annotation Error Detection methods can most effectively mitigate the impact of noisy data by detecting mislabeled instances? Datamap Confidence [30] proves to be an efficient approach at detecting errors in the dataset. It provided a precision and recall above 0.6, meaning that it can find over 60% of the errors having over a 60% of precision in their predictions. Then, this method can optimize significantly the error detection process, mitigating the impact of theose label errors.

5.3 Limitations and Future Work

In this work, we evaluated techniques that detect errors in the samples. We also estimated the error rates in the dataset. However, we disregarded this error rate in the comparison of the methods. Neither did we evaluate the impact of using the AED during the Active Learning process.

Future work also includes replicating this study on benchmark datasets, for further insights and better reproducibility, as Reuters-21578[10] and EUR-Lex[11].

[10] `https://archive.ics.uci.edu/ml/machine-learning-databases/reuters21578-mld/reuters21578.html`

[11] `https://data.europa.eu/data/datasets/eur-lex-statistics?locale=en`

Furthermore, the dataset from the Dark Web Monitor used in this work only makes use of the six high level labels, and disregards the lower-level labels. The use of these techniques in a bigger dataset, and with more labels also remains open and with that also the investigation of using continual learning approaches in this scenario. It would also be interesting to investigate the possibility of detecting changes over the classes over time. In other words, the definition of the abuse types might change with the time, and therefore, samples that at one moment belonged to one particular class, might now belong to another. This phenomena is known as concept shift. Future work includes exploring techniques to detect those changes in the class distributions over time, to find samples that might be mislabeled for the simple case of not being up-to-date.

6 Conclusions

This project explored an automated pipeline to add new data to a web page classification system for CFLW Cyber Strategies' Dark Web Monitor. Specifically, this project studied the possibility of combining AL, to select new data and AED, to identify noisy samples. To do so, we compared different state-of-the-art strategies, and evaluated their adequacy.

This study showed that AL strategies help to efficiently select the most important samples, making the model able to reach the same performance by querying fewer samples, reducing human labeling efforts. In this context, BADGE [3], CLUE [22] and Mean Max Loss provided the best performance, having Mean Max Loss a significantly shorter query time. Second, this work showed the necessity of having a robust model against noisy data, and provided a pipeline with AED techniques to find noise and reduce its impact. In this sense, Datamap Confidence [30] proved to be the best performing strategy.

In this work, we provided new insights into the use of AL and AED on a real-world dataset. To the best of our knowledge, this is the first study to do so on real-world data, combine AL and AED as well as compare many state-of-the-art approaches. The experiments show promising results in the use of AL to query new data and AED to identify mislabeled samples. We can, therefore, answer our third research question on AL and AED integration. We use Mean Max Loss as the AL strategy and Datamap Confidence [30] for AED. The resulting final algorithm for CFLW is described in Appendix J.

This work opens up possibilities for future work toward human-in-the-loop setups and shows the feasibility of combining different strategies for mitigating labeling efforts. In the future, we hope to include continual learning to also reduce the necessary computational resources to keep such a machine learning model up to date as new data is introduced to the system.

References

1. Cost-effective Batch-mode Multi-label Active Learning. Neurocomputing **463**, 355–367 (Nov 2021). https://doi.org/10.1016/j.neucom.2021.08.063, `https://www.`

sciencedirect.com/science/article/pii/S0925231221012534, publisher: Elsevier
2. Amiri, H., Miller, T., Savova, G.: Spotting Spurious Data with Neural Networks. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 2006–2016. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018). https://doi.org/10.18653/v1/N18-1182, https://aclanthology.org/N18-1182
3. Ash, J.T., Zhang, C., Krishnamurthy, A., Langford, J., Agarwal, A.: Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds (Feb 2020). https://doi.org/10.48550/arXiv.1906.03671, http://arxiv.org/abs/1906.03671, arXiv:1906.03671 [cs, stat]
4. Bernhardt, M., Castro, D.C., Tanno, R., Schwaighofer, A., Tezcan, K.C., Monteiro, M., Bannur, S., Lungren, M., Nori, A., Glocker, B., Alvarez-Valle, J., Oktay, O.: Active label cleaning for improved dataset quality under resource constraints. Nature Communications **13**(1), 1161 (Mar 2022). https://doi.org/10.1038/s41467-022-28818-3, http://arxiv.org/abs/2109.00574, arXiv:2109.00574 [cs]
5. Brinkhuijsen, S.: Context-Aware Feature Vectors in Dark Web Page Classification (2022)
6. Chakraborty, S., Balasubramanian, V., Panchanathan, S.: Optimal batch selection for active learning in multi-label classification. In: Proceedings of the 19th ACM international conference on Multimedia. pp. 1413–1416. ACM, Scottsdale Arizona USA (Nov 2011). https://doi.org/10.1145/2072298.2072028, https://dl.acm.org/doi/10.1145/2072298.2072028
7. Ein-Dor, L., Halfon, A., Gera, A., Shnarch, E., Dankin, L., Choshen, L., Danilevsky, M., Aharonov, R., Katz, Y., Slonim, N.: Active Learning for BERT: An Empirical Study. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 7949–7962. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.638, https://aclanthology.org/2020.emnlp-main.638
8. Gissin, D., Shalev-Shwartz, S.: Discriminative Active Learning (Jul 2019). https://doi.org/10.48550/arXiv.1907.06347, http://arxiv.org/abs/1907.06347, arXiv:1907.06347 [cs, stat]
9. Grivas, A., Alex, B., Grover, C., Tobin, R., Whiteley, W.: Not a cute stroke: Analysis of Rule- and Neural Network-based Information Extraction Systems for Brain Radiology Reports. In: Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis. pp. 24–37. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.louhi-1.4, https://aclanthology.org/2020.louhi-1.4
10. van Halteren, H.: The Detection of Inconsistency in Manually Tagged Text. In: Proceedings of the COLING-2000 Workshop on Linguistically Interpreted Corpora. pp. 48–55. International Committee on Computational Linguistics, Centre Universitaire, Luxembourg (Aug 2000), https://aclanthology.org/W00-1907
11. Huang, J., Child, R., Rao, V., Liu, H., Satheesh, S., Coates, A.: Active Learning for Speech Recognition: the Power of Gradients (Dec 2016). https://doi.org/10.48550/arXiv.1612.03226, http://arxiv.org/abs/1612.03226, arXiv:1612.03226 [cs, stat]
12. Huang, S.J., Jin, R., Zhou, Z.H.: Active Learning by Querying Informative and Representative Examples. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(10), 1936–1949 (Oct 2014).

https://doi.org/10.1109/TPAMI.2014.2307881, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence
13. Huang, S.J., Zhou, Z.H.: Active Query Driven by Uncertainty and Diversity for Incremental Multi-label Learning. 2013 IEEE 13th International Conference on Data Mining pp. 1079–1084 (Dec 2013). https://doi.org/10.1109/ICDM.2013.74, http://ieeexplore.ieee.org/document/6729601/, conference Name: 2013 IEEE International Conference on Data Mining (ICDM) ISBN: 9780769551081 Place: Dallas, TX, USA Publisher: IEEE
14. Karimi, D., Dou, H., Warfield, S.K., Gholipour, A.: Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis. Medical Image Analysis **65**, 101759 (Oct 2020). https://doi.org/10.1016/j.media.2020.101759, https://www.sciencedirect.com/science/article/pii/S1361841520301237
15. Kim, K.I.: Active Label Correction Using Robust Parameter Update and Entropy Propagation. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision – ECCV 2022. pp. 1–16. Lecture Notes in Computer Science, Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-19803-8$_1$
16. Klie, J.C., Webber, B., Gurevych, I.: Annotation Error Detection: Analyzing the Past and Present for a More Coherent Future (Sep 2022). https://doi.org/10.48550/arXiv.2206.02280, http://arxiv.org/abs/2206.02280, arXiv:2206.02280 [cs]
17. Larson, S., Mahendran, A., Lee, A., Kummerfeld, J.K., Hill, P., Laurenzano, M.A., Hauswald, J., Tang, L., Mars, J.: Outlier Detection for Improved Data Quality and Diversity in Dialog Systems (Apr 2019). https://doi.org/10.48550/arXiv.1904.03122, http://arxiv.org/abs/1904.03122, arXiv:1904.03122 [cs]
18. Li, S.Y., Shi, Y., Huang, S.J., Chen, S.: Improving deep label noise learning with dual active label correction. Machine Learning **111**(3), 1103–1124 (Mar 2022). https://doi.org/10.1007/s10994-021-06081-9, https://doi.org/10.1007/s10994-021-06081-9
19. Li, X., Guo, Y.: Active Learning with Multi-Label SVM Classification
20. Ma, X., Gao, J., Xu, C.: Active Universal Domain Adaptation. pp. 8968–8977 (2021), https://openaccess.thecvf.com/content/ICCV2021/html/Ma_Active_Universal_Domain_Adaptation_ICCV_2021_paper.html
21. Northcutt, C., Jiang, L., Chuang, I.: Confident Learning: Estimating Uncertainty in Dataset Labels. Journal of Artificial Intelligence Research **70**, 1373–1411 (Apr 2021). https://doi.org/10.1613/jair.1.12125, https://www.jair.org/index.php/jair/article/view/12125
22. Prabhu, V., Chandrasekaran, A., Saenko, K., Hoffman, J.: Active Domain Adaptation via Clustering Uncertainty-weighted Embeddings (Oct 2021). https://doi.org/10.48550/arXiv.2010.08666, http://arxiv.org/abs/2010.08666, arXiv:2010.08666 [cs]
23. Rebbapragada, U., Brodley, C.E., Sulla-Menashe, D., Friedl, M.A.: Active Label Correction. In: 2012 IEEE 12th International Conference on Data Mining. pp. 1080–1085 (Dec 2012). https://doi.org/10.1109/ICDM.2012.162, iSSN: 2374-8486
24. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks (Aug 2019). https://doi.org/10.48550/arXiv.1908.10084, http://arxiv.org/abs/1908.10084, arXiv:1908.10084 [cs]
25. Reyes, O., Morell, C., Ventura, S.: Effective active learning strategy for multi-label learning. Neurocomputing **273**, 494–508 (Jan 2018).

https://doi.org/10.1016/j.neucom.2017.08.001, https://www.sciencedirect.com/science/article/pii/S0925231217313371
26. Sener, O., Savarese, S.: Active Learning for Convolutional Neural Networks: A Core-Set Approach (Jun 2018). https://doi.org/10.48550/arXiv.1708.00489, http://arxiv.org/abs/1708.00489, arXiv:1708.00489 [cs, stat]
27. Settles, B.: Active Learning Literature Survey. Technical Report, University of Wisconsin-Madison Department of Computer Sciences (2009), https://minds.wisconsin.edu/handle/1793/60660, accepted: 2012-03-15T17:23:56Z
28. Song, H., Kim, M., Park, D., Shin, Y., Lee, J.G.: Learning from Noisy Labels with Deep Neural Networks: A Survey (Mar 2022). https://doi.org/10.48550/arXiv.2007.08199, http://arxiv.org/abs/2007.08199, arXiv:2007.08199 [cs, stat]
29. Su, J.C., Tsai, Y.H., Sohn, K., Liu, B., Maji, S., Chandraker, M.: Active Adversarial Domain Adaptation. In: 2020 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 728–737 (Mar 2020). https://doi.org/10.1109/WACV45572.2020.9093390, iSSN: 2642-9381
30. Swayamdipta, S., Schwartz, R., Lourie, N., Wang, Y., Hajishirzi, H., Smith, N.A., Choi, Y.: Dataset Cartography: Mapping and Diagnosing Datasets with Training Dynamics (Oct 2020). https://doi.org/10.48550/arXiv.2009.10795, http://arxiv.org/abs/2009.10795, arXiv:2009.10795 [cs]
31. Thompson, E.: MD5 collisions and the impact on computer forensics. Digital Investigation **2**(1), 36–40 (Feb 2005). https://doi.org/10.1016/j.diin.2005.01.004, https://www.sciencedirect.com/science/article/pii/S1742287605000058
32. Tor Metrics Team: Tor metrics (2023), https://metrics.torproject.org/
33. Wertz, L., Bogojeska, J., Mirylenka, K., Kuhn, J.: Evaluating Pre-Trained Sentence-BERT with Class Embeddings in Active Learning for Multi-Label Text Classification. In: Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). pp. 366–372. Association for Computational Linguistics, Online only (Nov 2022), https://aclanthology.org/2022.aacl-short.45
34. Wu, J., Sheng, V.S., Zhang, J., Li, H., Dadakova, T., Swisher, C.L., Cui, Z., Zhao, P.: Multi-Label Active Learning Algorithms for Image Classification: Overview and Future Promise. ACM Computing Surveys **53**(2), 28:1–28:35 (Mar 2020). https://doi.org/10.1145/3379504, https://doi.org/10.1145/3379504
35. Xie, B., Yuan, L., Li, S., Liu, C.H., Cheng, X., Wang, G.: Active Learning for Domain Adaptation: An Energy-Based Approach (Mar 2022), http://arxiv.org/abs/2112.01406, arXiv:2112.01406 [cs]
36. Yuan, M., Lin, H.T., Boyd-Graber, J.: Cold-start Active Learning through Self-supervised Language Modeling (Oct 2020). https://doi.org/10.48550/arXiv.2010.09535, http://arxiv.org/abs/2010.09535, arXiv:2010.09535 [cs]
37. Zhang, H., Zou, F.: A survey of the dark web and dark market research. In: 2020 IEEE 6th International Conference on Computer and Communications (ICCC). pp. 1694–1705 (2020). https://doi.org/10.1109/ICCC51575.2020.9345271

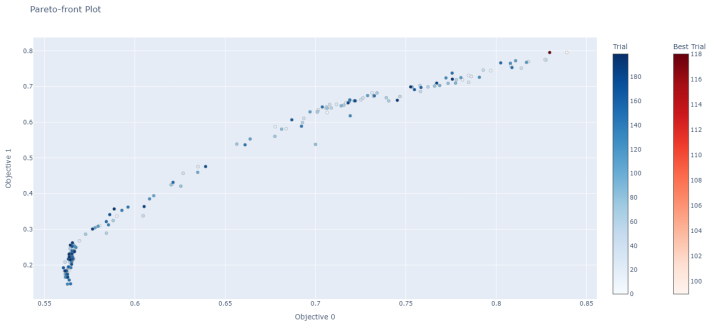


Fig. 4: Pareto's front of the hyperparameter optimization of the model. Objective 0 is Micro F1 and Objective 1 is Macro F1.

A Model characteristics

The classification model consists of a 2 layer Neural Network. The input size is 384, as it is the output size of the SBERT vectorization module. The hidden layer consists of a linear layer followed by a Dropout and a hyperbolic tangent. The last layer consists of a linear layer of size 6 (output size), followed by a sigmoid layer, as a multi-label classification is desired. Hyperparameters were optimized using Optuna[12], looking to optimize for Micro-F1 and Macro-F1. Figure shows the Pareto Front of this hyperparameter optimization. The chosen hyperparameters were: Hidden layer size 100, 100 training epochs with a batch size of 32, $5 * 10^{-3}$ learning rate, 0.2 dropout and weight decay. This model had an accuracy of 0.935, a Micro F1 of 0.832 and a Macro F1 of 0.797.

B Detailed explanation of DALUCS

Dual active learning based on Uncertainty and Cosine Similarity (DALUCS) is a novel Active Learning stratey that combinesFirst, DALUCS uses an uncertainty metric to detect examples with low predictive confidence. Inspired by CVIRS [25], DALUCS uses separation margin with Borda's ranking aggregation method. Hence, we start by computing the separation margin $m(i, l)$ for the predictions for each instance's labels:

$$m(i,l) = |P(\hat{y}_{i,l} = 1|x_i;\theta) - P(\hat{y}_{i,l} = 0|x_i;\theta)| \quad (1)$$

where x_i are the coordinates in the embedding space of each sample i, $P(\hat{y}_{i,l} = 1|x_i;\theta)$ is the probability of the model θ predicting class l for sample i. Given these margins, we create a vector of margins for each sample i. Now, we can compute a rank for each label, τ_l over all unlabeled samples.

[12] https://optuna.org/

To evaluate each instance over all labels, we aggregate their positions in each label ranking, τ_l, using Borda's method, which is computationally efficient [25]. The larger the value of $s_u(i)$, the more uncertain that instance is. This way, the uncertainty score for each instance is calculated as:

$$s_u(i) = \frac{\sum_l U_s - \tau_l(i)}{q(U_s - 1)} \tag{2}$$

where U_s is the number of unlabeled samples, q the number of labels and $\tau_l(i)$ the position of sample i in the ranking of label l. This model detects changes in the feature space by comparing features in the embedding space of the unlabeled samples with features of the labeled samples using cosine similarity. The aggregation of all similarities of unlabeled sample of the same class gives a similarity score s_s for that unlabeled instance:

$$s_s(i) = \sum_{x_l \in X_l} \frac{< x_i, x_l >}{||x_i||||x_l||} \tag{3}$$

where X_c represents the set of samples belonging to class l and x_c the coordinates in the embedding space of a particular sample in that set. After having the scores from all spaces, we will compute a sampler, one that takes into account uncertainty and label space. The sampler will give a score to each instance, where i^* are the selected samples to be labeled and β is a trade-off parameter between informativeness and feature domain exploration:

$$i^* = \arg\max_i \left(s_u(i)^\beta * (1/s_s(i))^{1-\beta}\right) \tag{4}$$

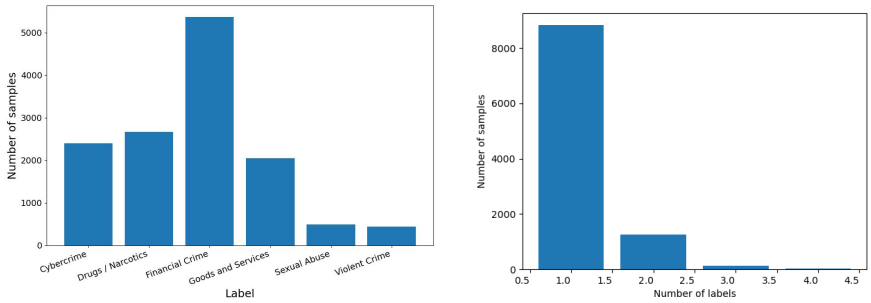
C Detailed explanation of DLED

DLED (Distance based Label Error Detection) is the flagger alternative to Mean Distance [17]. First, the mean points in the embedding space are calculated for each class, where X_l is the set of samples belonging to class l, x_l the coordinates in the embedding space of a particular sample in that set and n_l the number of samples belonging to class l:

$$mean_l = \frac{\sum_{x_l \in X_l} x_l}{n_l} \tag{5}$$

Next, for each sample i, its label correctness is assessed as follows. If the class of the sample corresponds to the closest mean, then the sample is considered to be correct. Otherwise, it is considered noisy.

$$Correctness_i = \begin{cases} 1, & \text{if } c = \arg\min_l(|x_i - mean|) \\ 0, & \text{otherwise} \end{cases} \tag{6}$$



(a) Number of samples containing each label.

(b) Number of labels per sample.

Fig. 5: Number of samples per label and number of labels per sample.

D Exploratory Data Analysis

Figure 5a depicts the data distribution across the different classes. It can be seen that there is a significant data imbalance. This was tackled by adjusting the Binary Cross Entropy function of the model to give more importance to the minority classes, reducing the effects of the data imbalance.

Figure 5b shows the number of labels per sample. Being this a multilabel classification problem, we can see that the majority of the samples only contain one label. Still, a significant amount of samples contains also 2 labels, while only a few of them contain 3 or 4 labels.

Figure 6 depicts the correlation between the different classes. Having the great majority of the classes just one label, we can see that in general there is a low correlation between them. However, it can be seen some correlation between classes "Sexual Abuse" and "Violent Crime", which suggests that the majority of the samples having more than one class belong to those classes.

E Active learning helps against data imbalance

Figure 7 depicts the distribution of label data at the start of the AL experiment, and at the end.

It can be seen that with Mean Max Loss, the data imbalance is slightly reduced. However, it is not enough by itself. One of the reasons is the lack of samples belonging to the minority classes. For example, there are 230 available samples belonging to class *Violent Crime*. With Mean Max Loss, all samples belonging to that class were sampled after half of the sampling rounds, after which it could not be augmented. On the contrary, random sampling was not able to sample all of them after all the rounds. Another reason is the sampling strategy. Mean Max Loss is not specifically designed to sample minority classes, but the samples that produce the highest loss on the model.

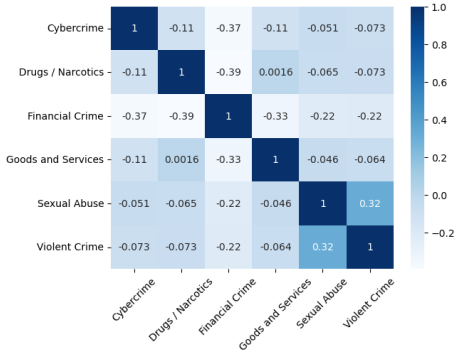


Fig. 6: Correlation between number of labels per sample.

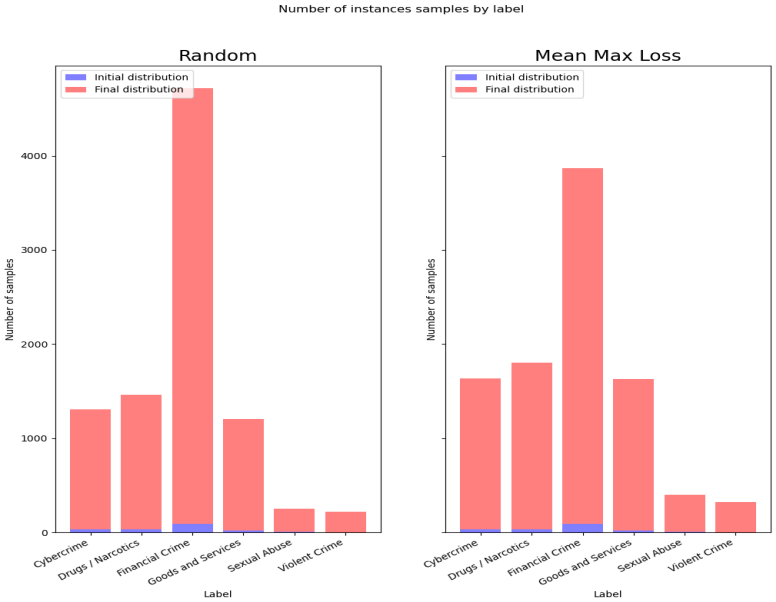


Fig. 7: Number of samples containing each label, before and after the sampling process using a) random sampling and b) BADGE [3].

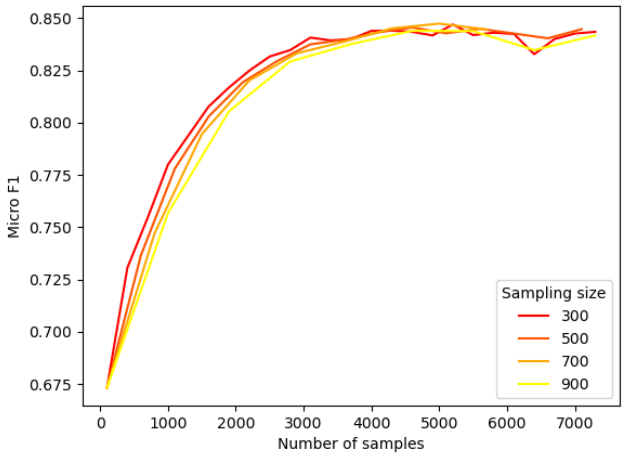


Fig. 8: Performance of the AL experiment depending on the number of samples

F Optimal sampling size

Figure 8 depicts the performance of the model depending on the number of samples. Different sampling size are compared in order to see if a specific one gives better performance for the same number of samples.

As it can be seen, lower sampling sizes give better performances in earlier stages of the AL process. However, as more data is available, the performances start to be more comparable, until there is no significant difference between the sampling sizes. For our specific purpose, we can say that there are no significant differences between sampling sizes, although there can be a slight preference for smaller batch sizes. Therefore, the sampling size should be decided in terms of the update frequency desired by CFLW, and the amount of newly incoming data.

G Additional results for active learning

In Figures 9, 10, and 11, we show the additonal results of accuracy and macro-F1, not reported in the main paper.

H Additional results for Annotation Error Detection

In Figures 12 (a) and 12 (b), we show the additional results of accuracy and macro-F1, not reported in the main paper.

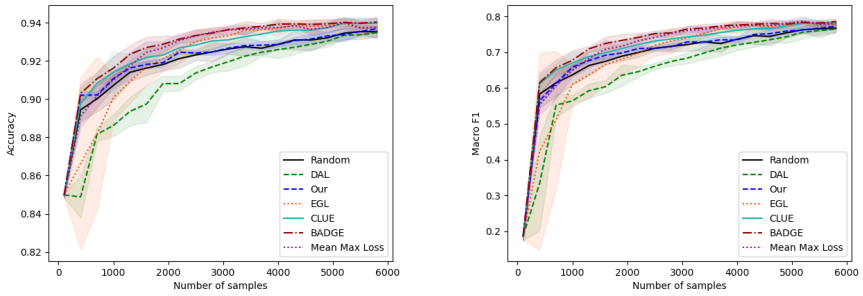


Fig. 9: Accuracy and macro-F1 of AL-normal.

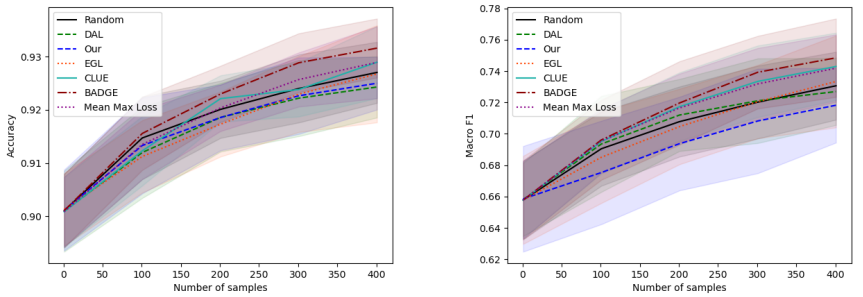


Fig. 10: Accuracy and macro-F1 of AL-slight shift.

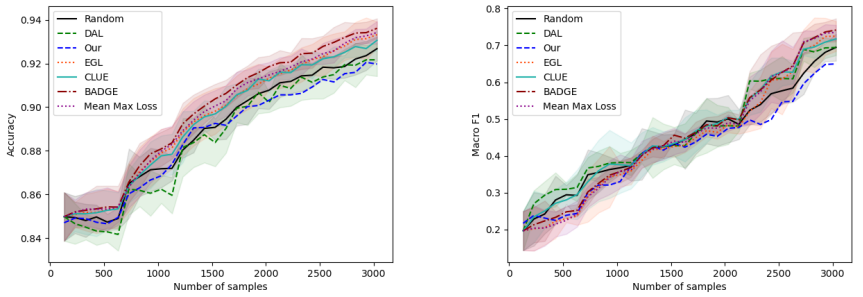
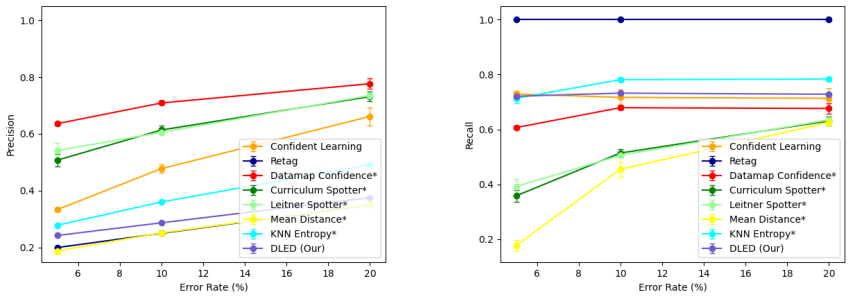


Fig. 11: Accuracy and macro-F1 of AL-extreme shift.



(a) Precision (b) Recall

Fig. 12: Comparison of different AED methods. X-axis describes the rate of the artificially introduced errors, y-axis shows performance.

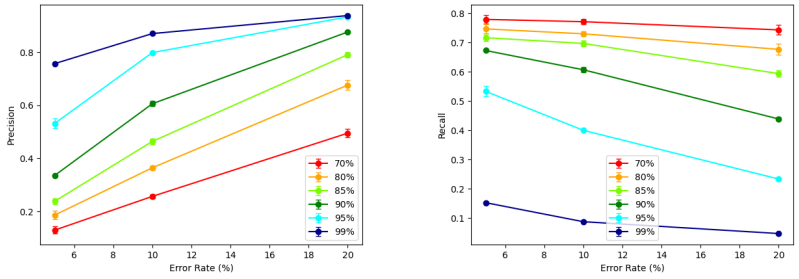
I Choosing the optimal threshold for annotation error detection scorer methods

Scorer methods provide to each sample a score corresponding to the likeliness of being erroneous. This score can vary greatly depending on the approach used, and is not always defined withing the same range. From this score, we can create a ranking defining the most probable mislabeled samples. However, this implies an extra difficulty when applying it in real setups. A threshold is necessary to cast the binary judgement between noisy and correct. In this work, it was decided that the most straightforward strategy was to select the top-n^{th} percentile. In this sense, in the literature usually scorer methods are evaluated by selecting the top 10% of the samples as mislabeled. However, this strategy might not be optimal. Figure 13 shows the performance of Datamap Confidence [30] as a function of the threshold chosen, represented as the n^{th} percentile. We can see that the higher the percentile, the higher the precision, but the lower the recall, and vice versa. In this sense, we are facing a precision-recall balance dilemma.

However, if we look at the F1-score, depicted in Figure 13, it can be observed that the optimal score is found by choosing the threshold as the percentile of correct samples. That is, if there is a 20% error rate in the dataset, there is an 80% of correct samples, then, the optimal threshold is represented by the 80% percentile of the scores. Nonetheless, this implies a following difficulty. In experimental setups, where there is a known error rate, this threshold is easy to choose. However, in real data the error rate is not known, therefore it needs to be estimated. This is one of the main motivation for the error estimation on our dataset.

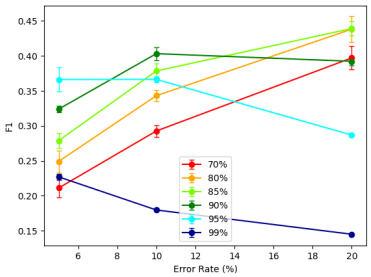
J Final Pipeline

Algorithm 1 explains the workflow of the final pipeline.



(a) Precision

(b) Recall



(c) F1-score

Fig. 13: Comparison of Datamap Confidence [30] performance as a function of the threshold chosen in terms of Precision and Recall.

Algorithm 1 pipeline of the proposed solution

Require: Labeled dataset X_l, Unlabeled dataset X_u, Test dataset X_t, predictive model with parameters θ, Sampling strategy S_l

for Number AL of steps **do**

$\theta \leftarrow train(model(\theta, X_l))$ ▷ Train model on labeled dataset

$results \leftarrow model(X_t)$ ▷ Store the model's performance

$X_s \leftarrow S_l(X_u)$ ▷ Query best instances according to Mean Max Loss

while Error rate > threshold **do**

$\hat{X}_e \leftarrow D_e(X_s)$ ▷ Detect noisy instances with Datamap Confidence [30]

$X_{new} \leftarrow review(\hat{X}_e)$ ▷ Human expert reviews instances with errors

end while

$X_l \leftarrow X_l \cup X_{new}$ ▷ Include selected and reviewed samples in the labeled dataset

$X_u \leftarrow X_u \setminus X_{new}$ ▷ Remove selected examples from the unlabeled dataset

end for