7 Detection of Emerging Cyberthreats Through Active Learning

Joel Brynielsson, Amanda Carp, and Agnes Tegen

7.1 INTRODUCTION

Machine learning holds great promise for detecting and responding to increasingly sophisticated cyberthreats [1]. The large volume of available data can, however, present challenges for effective data management. Annotating data typically requires a significant amount of time and resources, particularly if the task necessitates specialized knowledge. This raises the question of whether there are methods to reduce the manual labeling effort, while still achieving satisfactory performance. Active learning (AL) can reduce the necessary amount of labeled data by introducing human interaction in the training process. Through different query strategies, AL investigates the selection of data points by identifying the most informative samples to be labeled [2, 3].

This chapter, extending previous work [4], presents a study of the potential and application of AL to increase model performance for a binary text classification task. The aim is to fine-tune a transformer model for the purpose of classifying tweets to determine if an advanced persistent threat (APT) is mentioned. Incorporating AL in the training process seeks to avoid the laborious process of labeling data points that do not contribute further to spanning the outcome space. The main objective is to study which

DOI: 10.1201/9781003570882-9

AL approaches and strategies that are suitable for continuous improvement of identification of APTs in tweets. Hence, the research question studied is the following:

- What active learning approaches are effective for continuous improvement of classification of advanced persistent threats in tweets?

The remainder of this chapter is structured as follows. In Section 7.2, background to active learning is provided, along with how the technique can be related to the work of detecting cyberthreat actors. Section 7.3 then describes related work, followed by Section 7.4 discussing various strategies used to select data points for labeling, which is the central issue in active learning. Section 7.5 then outlines the method for the example of detecting cyberthreat actors in text fragments, which is explored in this chapter to illustrate the use of active learning for threat analysis. Section 7.6 presents the results of the conducted experiments in terms of how different strategies and parameters affect the machine learning performance. Section 7.7 includes a discussion of the results and experimental limitations to consider, followed by conclusions and recommendations for future work in Section 7.8.

7.2 BACKGROUND

This study focuses on the application of AL approaches for classifying tweets to identify potential threat actors within a cybersecurity context. Understanding the cyber perspective, the importance of identifying threat actors, and how it relates to continuous updating of a machine learning classifier, is therefore crucial for the purpose of this work.

7.2.1 Active Learning

AL has been used successfully in a wide variety of applications, such as computer vision [5, 6], activity recognition [7, 8], and natural language processing (NLP) [9]. AL aims to reduce the labeling effort required to train a model. Rather than labeling the entire dataset, only a small subset is labeled by querying an oracle. The oracle can be a human or a computer software. AL strategies choose which data points to label based on some type of informativeness criterion [10]. The goal is to select a representative set of labeled instances that capture the underlying distribution of the entire dataset. The performance of the model trained on this smaller set of labeled data can, with an optimal selection of data points, be comparable to a model trained on a much larger labeled dataset [2]. In most work on AL, it is assumed that the oracle is always correct and acts in accordance with what is expected, but this is not always the case [11, 12]. It is important to consider whether the assumptions about the oracle are reasonable, given the application at hand, for example contemplating whether the oracle is an expert or not.

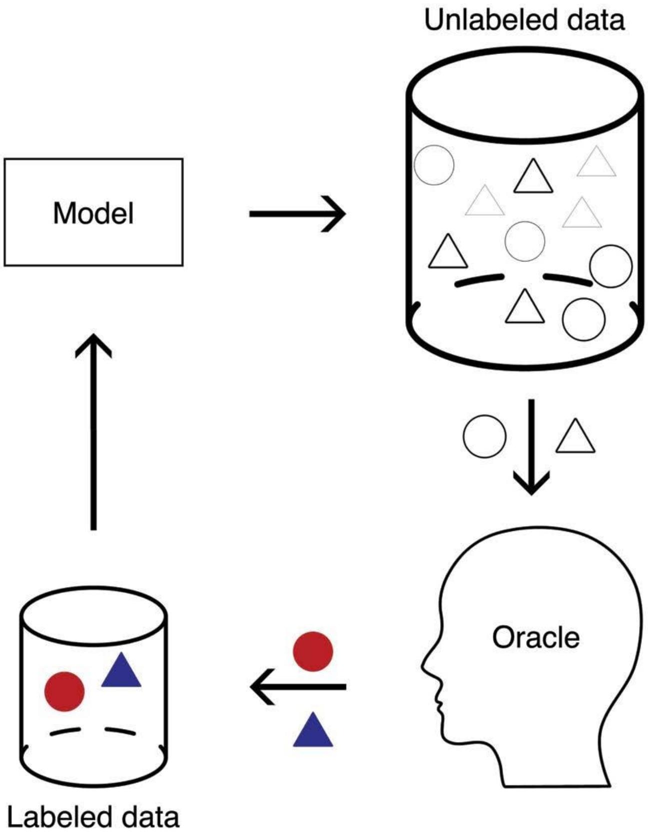


FIGURE 7.1 Active learning cycle.

Figure 7.1 illustrates an iterative training process of active learning, which is commonly employed. The training process is repeated until the model performs robustly, the labeling budget is used, or certain criteria are met. The labeling budget is often a percentage of the total amount of data [13]. The choice of AL strategy is dependent on the problem at hand. It can be difficult to estimate whether one strategy performs better than another before they have been put to the test. Settles [14] states that random sampling, which typically is used as a baseline, might be advisable if the problem is not well understood.

7.2.2 The Cyberthreat

Traditional cyberthreat detection methods rely on preventive work and network monitoring [15]. However, identifying and remediating cyberattacks take time. As threat actors grow more complex, new complementary technologies and methods are needed to identify and counter cyberattacks [16]. Cyber actors use social media, open forums, and the darknet to plan attacks, and the results of attacks are often sold or exposed online. Analyzing unstructured data from open sources can thus assist in predicting cyberattacks and cyberthreats.

APT is a term that is used to label a specific type of threat actor. An APT is usually a particularly well-resourced, stealthy adversary who is able to target specific information, and also eventually acquire it through persistent efforts [17]. The APT will typically succeed even if the target is a competent high-profile company or even a government. APTs are typically conducting long-term campaigns that involve multiple stages, utilizing the full range of their capabilities. According to the U.S. National Institute of Standards and Technology [18], APTs demonstrate a high level of expertise while they also possess large amounts of resources, enabling them to leverage multiple attack vectors, such as cyber, physical, and deceptive tactics. These attacks primarily involve infiltrating the targeted entity's information technology infrastructure to gain confidential information, disrupt vital aspects of a mission or organization, or position themselves to achieve similar objectives in the future.

APTs are typically given a name or a number by the first organization that discovers and publishes findings about them. However, these organizations, often antivirus and other types of cybersecurity companies, normally use their own naming conventions for an APT, regardless of who named it first [19]. This can lead to serious confusion. APT28, for example, has multiple aliases, such as Fancy Bear, Strontium, Pawn Storm, Sofacy, Sednit, and Tsar Team [19]. APT28, mentioned here as an example of an active APT, is a Russian-associated group that has been extensively documented and analyzed due to its involvement in multiple high-profile cyberattacks. The group has a long history of performing attacks with the common goal of promoting the political interests of the Russian government.

Cyber intelligence analysts have various roles. Some seek to assess the various APTs' capabilities to make threat assessments by analyzing and evaluating computer networks and systems [15]. They typically use various tools and techniques to monitor network traffic and activity, to detect patterns or anomalies that may indicate a cyberattack or a security breach. Actions to prevent or mitigate cyberattacks can then take place at different levels. At the strategic level, long-term measures are required, for example, replacing an entire system, or overhauling an architecture, due to an excessive number of security risks [20]. At the tactical level, responses are often more time-sensitive. Associated necessary measures should be implemented more swiftly, which may include, for example, updates of firewall rules or changes in routing tables.

Intelligence analysts possess considerable expertise in identifying and recognizing APTs. As such, they are potential users of the outcome of the study presented in this chapter, where the intelligence analysts fulfill the role of labeling data points. Through this process, the analysts can make valuable contributions to the training of the system through AL approaches, without necessarily having to share secret data with a system designer. This, in turn, secures that the system continues to stay pertinent, while accommodating additional data.

7.3 RELATED WORK

Several surveys explore AL within NLP applications. Olsson [9] presents an overview of the area, especially focusing on the theory and methodology that different AL approaches use for data selection. Much of the content can be generalized to AL in other applications as well, and is not specific to NLP. Miller et al. [21] present an overview, as well as simulation studies to investigate performance, efficiency, and practical applicability. They use support vector machines (SVMs) with data from Twitter, Wikipedia talk pages, and news articles in their experiments. Margin sampling, that is, uncertainty sampling based on the distance from the data points to the SVM hyperplanes, performs the best in their experiments. They also find that the length and style of the text data affect the results. In Wang et al.'s [22] study, human-in-the-loop NLP frameworks are discussed from both the machine learning perspective and the human-computer interaction perspective. They classify the surveyed papers in terms of task, goal, human interaction, and feedback learning method. Zhang et al. [23] showcase how the number of publications focusing on AL in the ACL Anthology[1] has increased over the last 15 years, indicating an increased interest in the subject. They discuss the current status of AL in NLP, as well as suggested future directions. Stiennon et al. [24] introduce a human feedback model for producing summaries of text data. In experiments with data from Reddit, they show that their model improves the quality of summaries, compared to supervised learning. Zhang et al. [25] study how active learning can aid in the fine-tuning of large language models (LLMs). LLMs, like deep learning in general, need a large amount of data to be trained. The paper introduces an approach where AL is combined with existing fine-tuning techniques to improve data efficiency. Experiments show that the new approach is better than baseline models on three complex reasoning tasks. Hu et al. [26] explore how AL together with LLMs handle code-related tasks. They study 11 different sampling strategies with three different LLMs, and find that classification-related tasks yield good performance, while nonclassification tasks do not yield as good results. In their experiments, they also find that clustering-based strategies outperform uncertainty-based strategies.

In the work mentioned above, AL within different NLP applications is studied, but not explicitly with regard to the cybersecurity domain. Bhattacharjee et al. [27] study the task of classifying phishing or malicious URLs. They propose an uncertainty-based AL strategy to help with the task, and experiments show an increase in the results. Lin et al. [28] investigate how malicious mislabeling and data poisoning attacks can affect AL of deep neural networks. They propose a clustering-based strategy and perform experiments on an image dataset. The results show that the suggested AL strategy is robust against mislabeling and data poisoning attacks. Moskal and Yang [29] introduce an approach that combines AL, transfer learning, and pseudolabels to aid analysts in interpreting possible intrusion alerts. They use a minimal amount of data, but still yield significant results. Pal et al. [30] take the perspective of the attacker, instead of defending against attacks. They focus on model extraction, where the aim of the attacker is to be able to replicate an unknown model, without access to the training data or knowledge about the employed base model. Their framework, denoted ActiveThief, is able to extract models in a variety of domains, from image to text. They compare different AL strategies within the framework and find that they get better results compared to the baseline random strategy.

Li et al. [31], Srivastava et al. [32], and Xie et al. [33] all study how AL can be used in named entity recognition (NER) tasks. Deep neural network models have resulted in good performance, but are typically dependent on the amount of data, which is why AL can be a possible solution. Li et al. [31] present an adversarial AL framework to pick the most informative samples for annotation. Their AL strategy is based on comparing the similarity between unlabeled samples and the already-labeled samples. Srivastava et al. [32] combine reinforcement learning with AL in their proposed method. Xie et al. [33] focus specifically on Chinese NER, which has not been studied much and is a complex task. They introduce a strategy combining uncertainty, confidence, and diversity.

Compared to previous work as discussed, this study focuses on evaluating the performance of different AL strategies in classifying APTs in tweets. A new dataset is used and the case when annotated data is scarce is studied specifically. The scarcity of data is motivated, as previously mentioned, by the expensive process of annotating data, especially when expert knowledge is needed.

7.4 ACTIVE LEARNING STRATEGIES

The central research question in active learning concerns how data points for annotation are selected. An example of how the selection of data points can be performed using two different strategies is illustrated in Figure 7.2. Four AL strategies are studied in the experiments, including the random strategy. The random query strategy (AL-random) selects data points randomly for labeling, and is used as a baseline for comparison with the other strategies. The other strategies are uncertainty sampling with entropy for uncertainty measurement (AL-entropy), diversity sampling using K-means (AL-kmeans), and cost-effective active learning (CEAL-entropy).

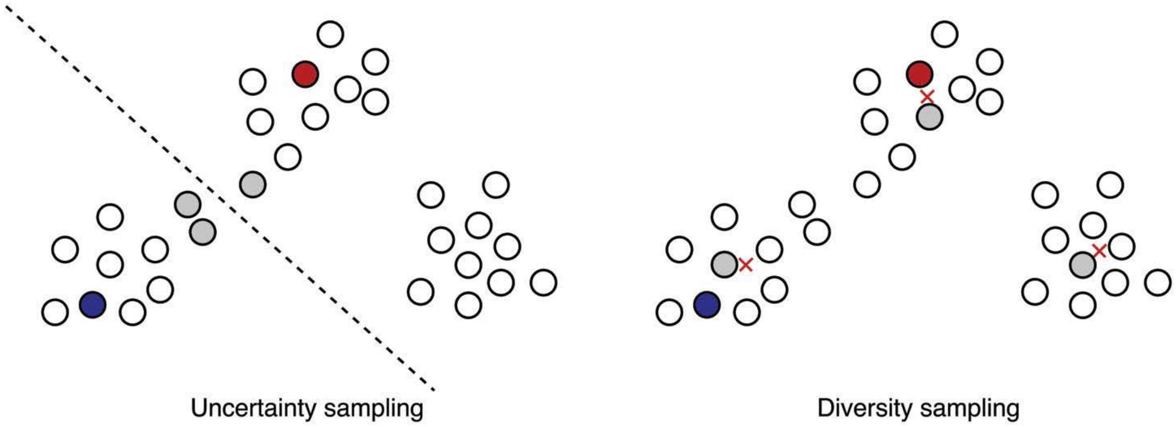


FIGURE 7.2 Data point selection by different AL query strategies. Red and blue points indicate labeled samples for two classes, and gray points represent samples selected for oracle labeling for the respective query strategy. The dashed line marks the calculated dividing line between the two classes, and the red crosses mark the centroids of the clusters.

7.4.1 Uncertainty Sampling

Uncertainty sampling is based on selecting the samples that the model is most uncertain about how to classify. Thus, instances where the model is highly uncertain are supposed to be maximally informative [2]. A common approach to evaluate the predictions made by a model is to assess the probabilistic distribution of the classes. There are several uncertainty-based query strategies to measure this, such as least confidence, margin of confidence, and entropy. Entropy, a measure of impurity of a system [34], is widely used in machine learning as a measure of uncertainty of a model. The higher the entropy value, the more uncertain the model is about which class the data point belongs to. Hence, for binary classification, entropy-based sampling is the same as choosing the data point with posterior closest to 0.5.

7.4.2 Diversity Sampling

Uncertainty sampling is prone to selecting outliers and data that may not accurately represent the dataset [35]. Conversely, diversity sampling mitigates these concerns by identifying a subset of samples that comprehensively cover the entire dataset. Depending on the methodology employed to construct the subset, there exists a variety of diversity sampling techniques. One technique is cluster-based sampling, which is a method used to find structures among the unlabeled data points, where a commonly used strategy is K-means. This involves clustering a set of samples into K clusters, each characterized by a centroid point that minimizes the associated inertia [36].

7.4.3 Cost-Effective Active Learning

In addition to only selecting data points that the model is least confident about, cost-effective active learning (CEAL also considers samples where the model is most confident [37]. For instance, if the model predicts a data point belonging to a class with certainty 0.5, it is a likely candidate for uncertainty sampling. However, if the prediction is 1, it can be inferred that the model is maximally confident in its classification. In each AL cycle, the CEAL technique selects samples at both extremes: those with the highest uncertainty, and those with the lowest. For the latter, CEAL suggests provisionally labeling them based on the model's predictions, creating so-called pseudolabeled samples [37]. Subsequently, both the pseudolabeled samples and the oracle-labeled data points are added to the labeled training dataset, which is used to train a new model. Upon completing the training of the new model, the pseudolabeled samples are eliminated from the training dataset, and a new CEAL cycle is initiated. This process is depicted in Figure 7.3.

The unlabeled samples with an uncertainty measurement below a predetermined threshold δ are considered the most certain. The threshold for high-confidence sample

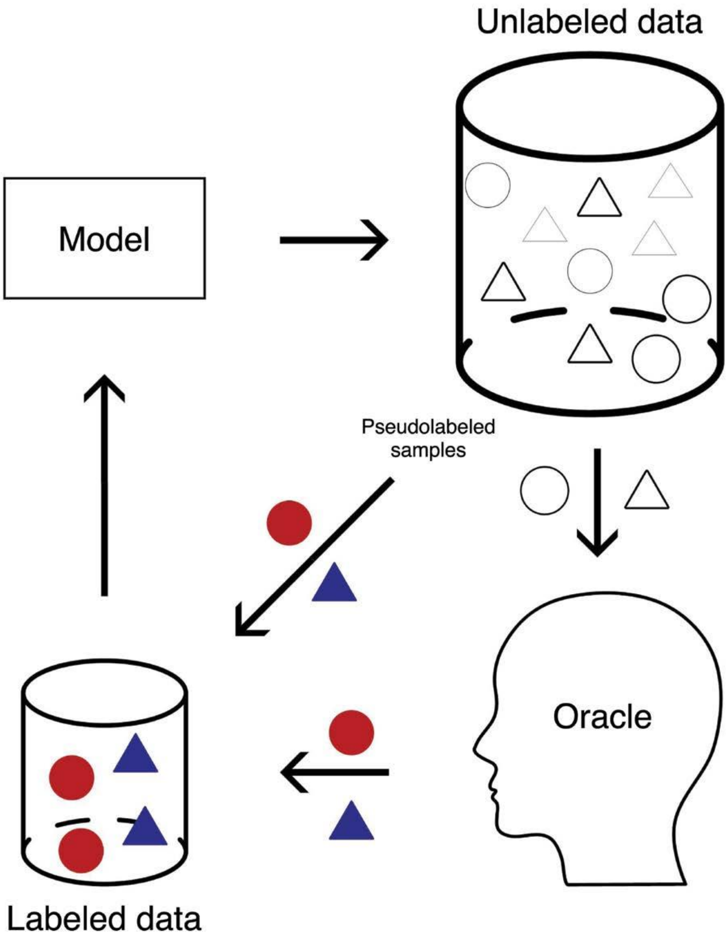


FIGURE 7.3 Cost-effective active learning.

selection is updated at each epoch, according to Equation 7.1. This is to be done to ensure that the labeling process remains dependable [37]. The threshold δ is defined by:

$$\delta = \begin{cases} \delta_0, & \text{for } t = 0, \\ \delta - dr \times t, & \text{for } t > 0, \end{cases} \tag{7.1}$$

where δ_0 is the initial threshold, dr controls the threshold decay rate, and t is the current epoch.

7.5 METHOD

This section describes the model used, the data, and the experimental setup. Figure 7.4 displays the iterative process used to train and evaluate the model. The chosen AL strategy picks data points from the pool of unlabeled data based on the defined selection criteria. The chosen data points are presented to the oracle, who in turn annotates them. The now-annotated data points are added to the annotated

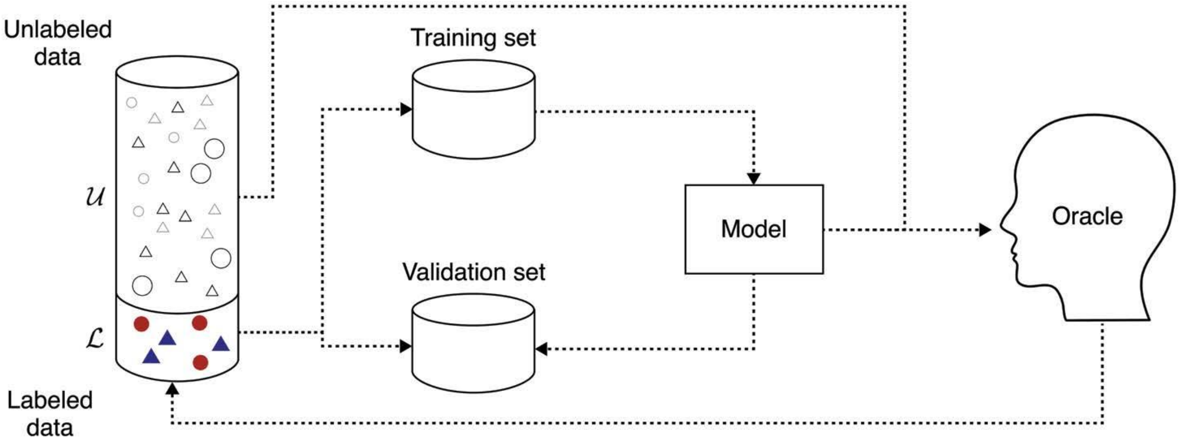


FIGURE 7.4 The iterative AL training and validation process employed in the experiments.

data, which in turn is used as a training set to retrain the model. The updated model is tested on the validation set and a performance score is received. After this final step, the cycle restarts and the AL strategy picks new data points for the oracle to annotate.

7.5.1 DistilBERT for Text Classification

DistilBERT is an open-source NLP framework used for the text classification part of the experiments. DistilBERT is smaller, faster, cheaper, and lighter than its predecessor BERT [38]. By using a small model, the time and resource costs associated with model training can be reduced while still maintaining high performance. The pretrained transformer DistilBERT, as described, was used through the Hugging Face library [39]. The tweets were tokenized and [CLS] and [SEP] tokens, used for classification and sentence separation, were added. The [CLS] token captures the entire context of the input for simple downstream tasks, such as classification. For sentence representations used in classification tasks, the size of the [CLS] token is equal to the number of data points × the number of hidden states. The tokenized input was padded to match the length of the longest tweet in the dataset. An attention mask was also created to distinguish the padded tokens from the nonpadded ones. The stochastic optimizer Adam [40] was utilized. A small search was conducted to identify an optimal learning rate for this classification task. Various learning rates were tested, focusing on values near the suggested learning rates mentioned for the original BERT model [41]. The search resulted in an optimal learning rate of $2e - 5$. A single linear layer was added at the output hidden state of the [CLS] token, on top of the DistilBERT model, to perform classification.

The pretrained model and the additional untrained classification layer were trained and updated at every iteration for the specific task. The cross-entropy loss was used to measure the performance of the model, calculated by comparing the divergence between the predicted probability and the actual label.

7.5.2 Dataset

The dataset used in the experiments contains approximately 35,000 labeled tweets [42]. Cyber-related tweets were identified by their association with keywords such as “cyber” and “malware.” An existing infrastructure for data download and rule-based detection of known APTs was leveraged to download large amounts of cyber-related tweets and automatically categorize them into two groups: texts with and without (known) APTs.

The dataset contains a total of 70 different APTs. A language detector from the fastText [43] library was utilized to identify and discard all tweets where English was not the most probable language. To enhance the model in locating APTs, distracting elements in all tweets were eliminated. Links, email addresses, phone numbers, and usernames were replaced with their respective masking tokens (“LINK,” “MAIL,” “PHONE,” and “USER”). Emojis were then converted to descriptive ones (for example, “👍” was changed to “:thumbs up:”) using the demoji Python package.[2] Duplicate tweets were removed, and the tweets were also normalized, for example, replacing “a . m .” and “p . m .” with “a.m.” and “p.m.”

Approximately 19,000 tweets remained after the cleaning. The entire dataset contained twice as many tweets belonging to the negative class as the positive class. To prevent the model from overtraining on a small number of negative examples, a skewed distribution with three times more negative examples was chosen for the training. An even distribution between positive and negative was chosen for the validation set. For the unlabeled pool dataset, the remaining data was added, resulting in 66% negative samples. For clarification, this is presented in Table 7.1.

Figure 7.5 displays the final preprocessed dataset, with the x-axis representing the length of tweets, and the y-axis representing the number of tweets, starting with the first bar representing one-word length.

7.5.3 Experimental Setup

In Algorithm 1, the main loop of the experiment is documented using pseudocode. The algorithm shows that the total number of data points added to the training dataset $\mathcal{L}$ is $K \times N$, where K is the number of samples in a batch and N is the number of epochs. The F-score and the accuracy were accumulated over all batches and logged at each epoch for the validation dataset. To obtain a fair evaluation and comparison between AL approaches, the training was averaged over three runs with 10

TABLE 7.1 Data distribution per class

DATASET	*POSITIVE*	*NEGATIVE*
Training	25%	75%
Validation	50%	50%
Unlabeled pool	34%	66%

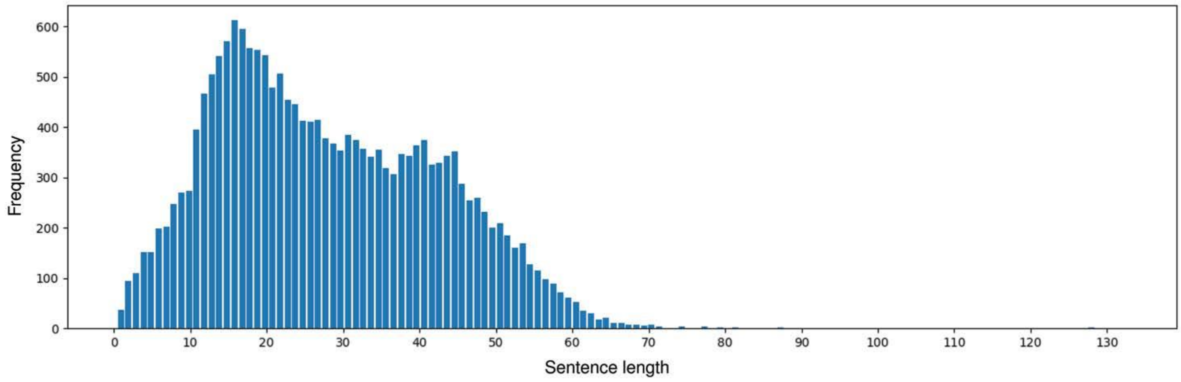


FIGURE 7.5 The frequency of sentence lengths in the final preprocessed dataset.

different seeds (101, 102, …, 110). The stopping criterion for training was when the maximum number of epochs was achieved. At every iteration, the model $\mathcal{M}$ is fine-tuned and thereby updated. At the end of every epoch, data points chosen according to a query strategy are added to the training dataset. In experiments involving the CEAL approach, the training dataset is augmented with pseudosamples during the fine-tuning of the model, and after the fine-tuning these pseudosamples are then returned to the unlabeled data pool before the next iteration, as described in Section 7.4.3.

Algorithm 1 Active learning experiment design

Inputs: Pretrained transformer model $\mathcal{M}$, unlabeled pool dataset $\mathcal{U}$, initially labeled dataset $\mathcal{L}$, validation dataset $\mathcal{V}$, acquisition size K for AL sampling, threshold δ for CEAL pseudosample inclusion (0 if CEAL is not to be used), maximum number of epochs N.

Output: Fine-tuned model $\mathcal{M}$.

```
for $i = 0, 1, \ldots, N$ do
    Fine-tune $\mathcal{M}$ with $\mathcal{L}$.
    Evaluate $\mathcal{M}$ on $\mathcal{V}$ and log results.
    Move back any pseudosamples from $\mathcal{L}$ into $\mathcal{U}$.
    Move $K$ samples from $\mathcal{U}$ into $\mathcal{L}$ based on a query strategy.
    Move pseudosamples with uncertainty below $\delta$ from $\mathcal{U}$ into $\mathcal{L}$.
    Update $\delta$ according to Equation 7.1.
end for
```

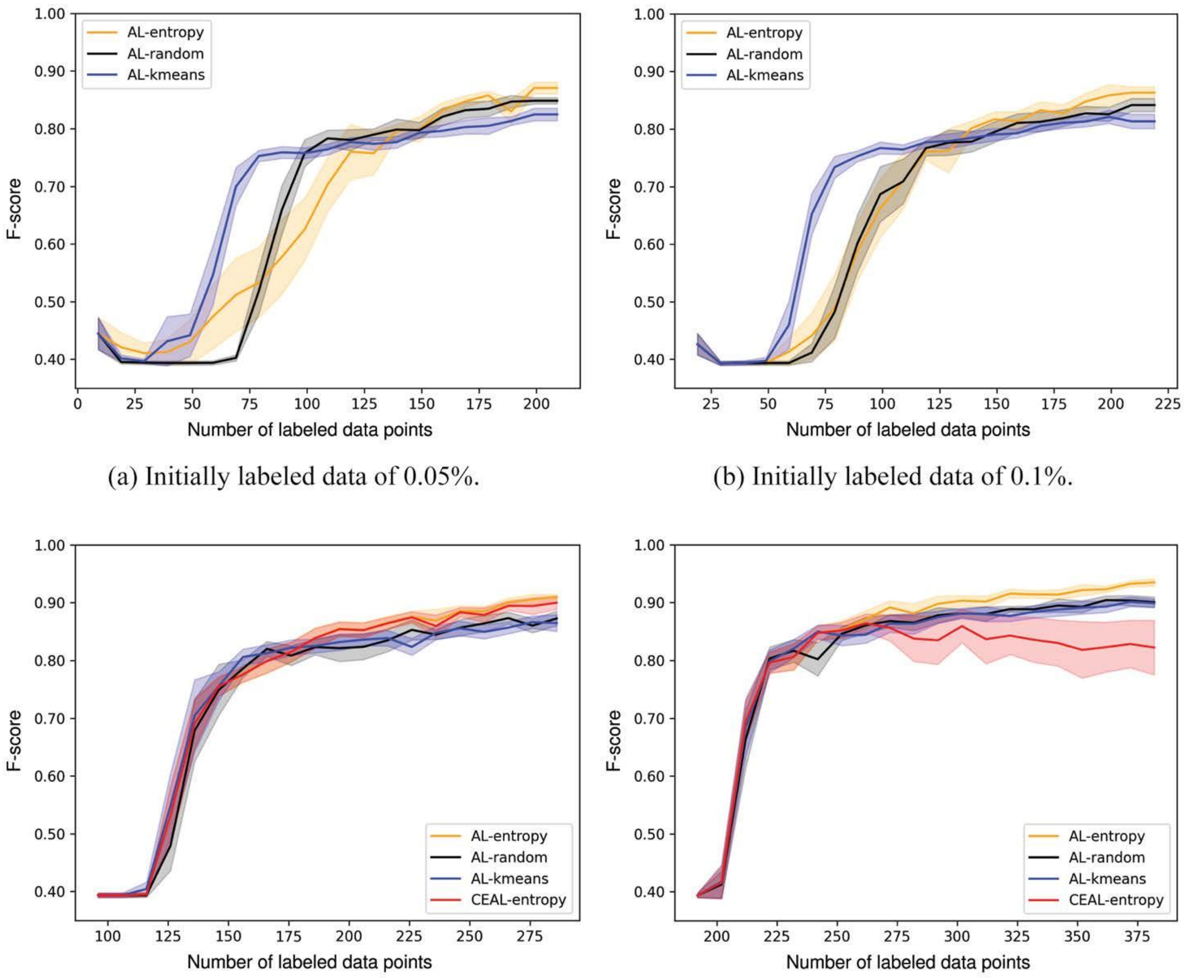
For diversity sampling, K-means clustering was used to sample diverse data points, deviating from uncertainty sampling where entropy was based on probabilities of the different classes. K-means was performed on the [CLS] token, which is a special classification token corresponding to the last hidden state in the DistilBERT model. K data points were then chosen to be sent to an oracle for labeling; for diversity sampling with K-means, the K data points were based on the smallest distance to each centroid, and for uncertainty sampling, the *K* most uncertain data points according to the classification by the currently fine-tuned model were chosen.

The CEAL approach required optimal values for the initial threshold δ_0 and the decay rate *dr* to be set in order to be implemented. The value δ sets the limit for the number of samples that are transferred to the labeled training dataset, and *dr* determines the rate of decay of δ over the number of epochs, as described by Equation 7.1. The decay rate *dr* was chosen to be 0.0033, as stated as the most optimal value according to the literature [37]. An initial threshold δ_0 of 0.35 was established through experimentation. The threshold allowed for the addition of pseudolabeled samples, that is, data points with entropy lower than the threshold are included in the training dataset. The addition of the pseudosamples had the potential of improving performance. Yet there was also a risk of decreased performance if incorrect labels were assigned.

The amount of initially labeled data and the acquisition size were varied to determine their impact on the model's performance. The amount of initially labeled data refers to the data used to train the model at the start of the experiment, and the acquisition size refers to the number of data points added each epoch. For the amount of initially labeled data, experiments were conducted with 0.05%, 0.1%, 0.5%, and 1% of the whole dataset, corresponding to 9, 19, 96, and 192 data points, respectively. The acquisition sizes tested in the experiments were 10, 25, and 50 data points. The validation set was set to be 4% of the whole dataset. To prevent misleading results due to lack of data in the validation set, a consistent amount of data was allocated for validation, regardless of the size of the training set. The conducted experiments were executed on a high-performance NVIDIA DGX A100 computing cluster consisting of eight NVIDIA A100 40 GB Tensor Core GPUs.

7.6 RESULTS

The four different query strategies are referred to as AL-entropy (uncertainty-based sampling), AL-random (random sampling used as baseline), AL-kmeans (diversity-based sampling), and CEAL-entropy (the CEAL strategy). The presented results referred to as CEAL-entropy are solely based on AL-entropy+CEAL-entropy. All combinations, that is, AL-entropy+CEAL-entropy, AL-random+CEAL-entropy, and AL-kmeans+CEAL-entropy, were tested, but due to their similar performance and space constraints, not all combinations are shown. As stated, the incorporation of pseudolabeled samples depended on the model's classification confidence to identify data points suitable for inclusion in the training dataset. Consequently, the results for CEAL-entropy are presented only for experiments in which the model displayed sufficient confidence in classifying data points. In the graphs presented in Figures 7.6–7.8, the *x*-axis denotes the number of labeled data points by the oracle, not the pseudolabeled samples.



(a) Initially labeled data of 0.05%.

(b) Initially labeled data of 0.1%.

(c) Initially labeled data of 0.5%.

(d) Initially labeled data of 1%.

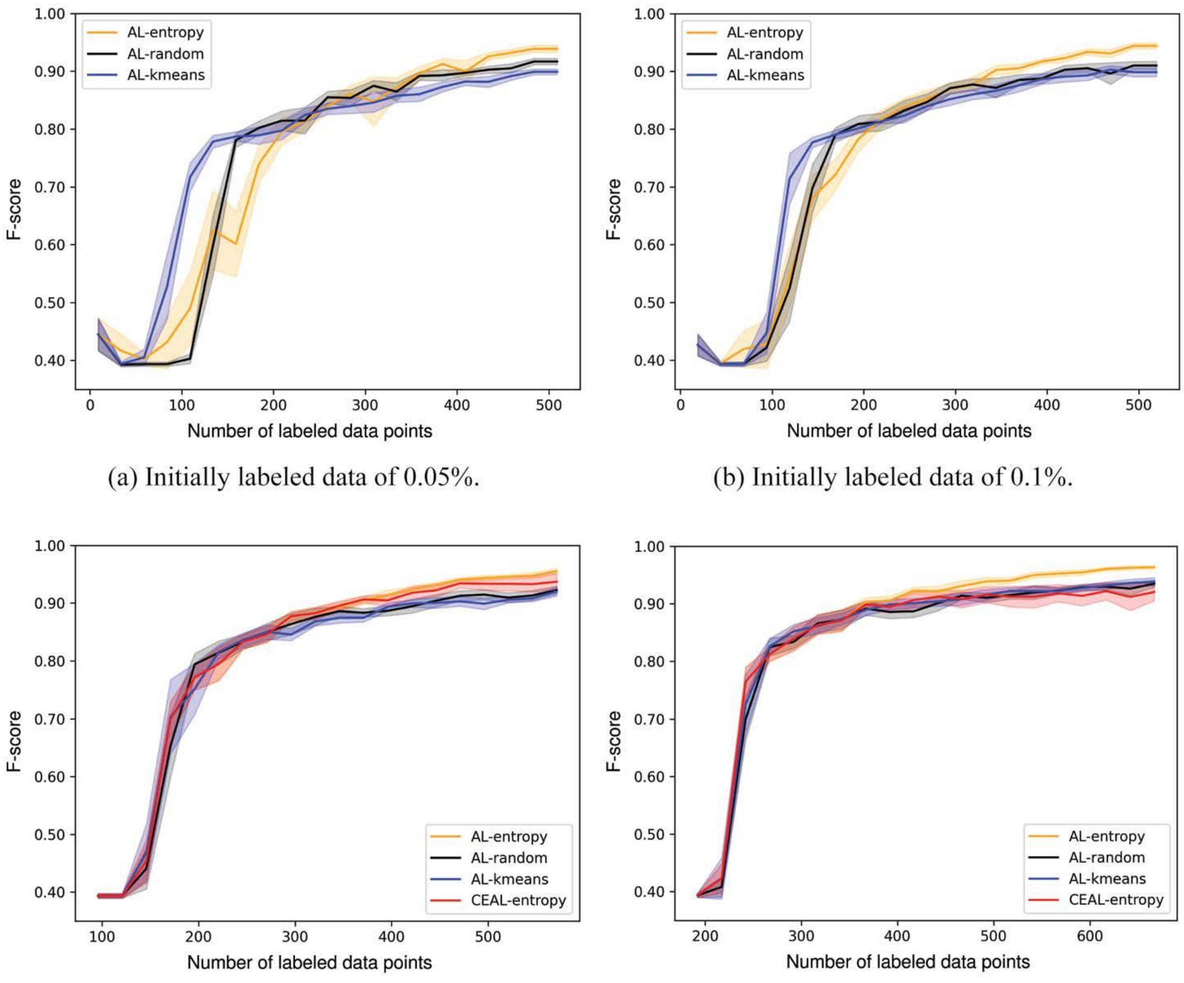
FIGURE 7.6 Average F-score of AL approaches and query strategies with different amounts of initially labeled data and acquisition size 10, averaged across 10 seeds and shown with 95% confidence intervals.

7.6.1 Acquisition Size 10

In Figure 7.6, four graphs are presented displaying the F-score of four scenarios with acquisition size 10 and different quantities of labeled data that the model had at its disposal for training. The experimental setup involved conducting experiments with an acquisition size of 10 over a span of 20 epochs, with varying amounts of initially labeled data. The total number of labeled data points was determined by adding the initially labeled data to the 200 data points (10 × 20) acquired from the pool of unlabeled data. This was the procedure for all query strategies, except when employing CEAL.

7.6.2 Acquisition Size 25

Figure 7.7 contains four graphs representing the F-score across four scenarios with acquisition size 25 and varying quantities of labeled data available for model training.



(a) Initially labeled data of 0.05%.

(b) Initially labeled data of 0.1%.

(c) Initially labeled data of 0.5%.

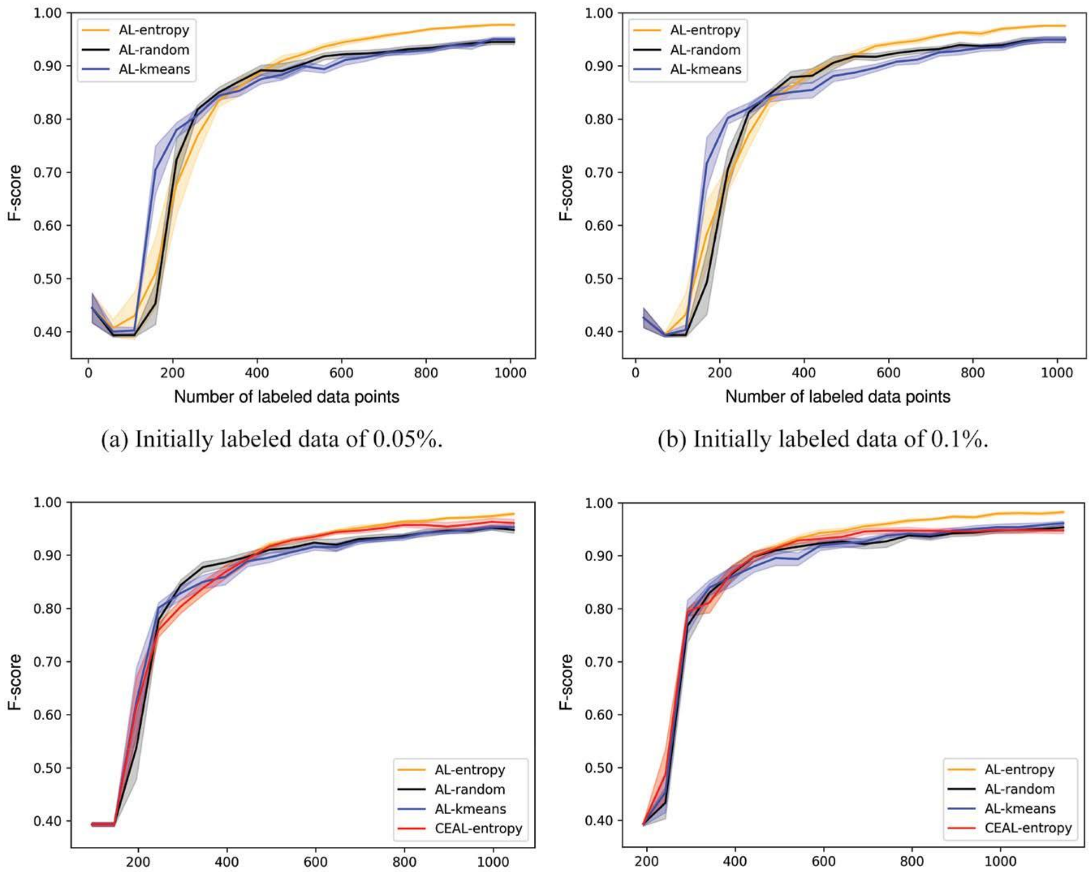
(d) Initially labeled data of 1%.

FIGURE 7.7 Average F-score of AL approaches and query strategies with different amounts of initially labeled data and acquisition size 25, averaged across 10 seeds and shown with 95% confidence intervals.

The experiments were conducted across 20 epochs, with varying amounts of initially labeled data. Similar to the previous section, the total number of labeled data points was determined by adding the initially labeled data to the 500 data points (25 × 20) acquired from the pool of unlabeled data.

7.6.3 Acquisition Size 50

In Figure 7.8, the F-scores for four separate scenarios with acquisition size 50 are displayed, characterized by the varying quantities of labeled data available to the model during the training process. As in the previous sections, the experiments were carried out over 20 epochs, with different amounts of initially labeled data. The total number of labeled data points was calculated by adding the initially labeled data to the 1,000 data points (50 × 20) acquired from the pool of unlabeled data, to provide a comprehensive understanding of the performance trends under this experimental condition.



(a) Initially labeled data of 0.05%.

(b) Initially labeled data of 0.1%.

(c) Initially labeled data of 0.5%.

(d) Initially labeled data of 1%.

FIGURE 7.8 Average F-score of AL approaches and query strategies with different amounts of initially labeled data and acquisition size 50, averaged across 10 seeds and shown with 95% confidence intervals.

7.6.4 Comparison of Acquisition Sizes

Table 7.2 presents a comparison between acquisition sizes 10 and 50, using 0.5% of the initially labeled data (96 data points). It presents the effects of acquisition sizes on

TABLE 7.2 F-score for acquisition sizes 10 and 50, when trained on the same number of data points

DATA POINTS	*ACQ. SIZE*	*EPOCHS*	*AL-ENTROPY*	*AL-RANDOM*	*AL-KMEANS*	*CEAL-ENTROPY*
196	10	6	0.76	0.75	0.76	0.76
	50	2	0.39	0.39	0.39	0.39
246	10	11	0.86	0.82	0.83	0.85
	50	3	0.61	0.54	0.62	0.61
296	10	16	0.89	0.86	0.86	0.88
	50	4	0.76	0.78	0.80	0.76

performance, with the F-score for each of the tested query strategies given for a specific number of data points, thus resulting in different numbers of epochs. To compare how the F-score is affected by the same number of data points with varying acquisition sizes, the rows should be analyzed in pairs.

7.7 DISCUSSION

As can be seen in Figures 7.6–7.8, each of the tested query strategies yielded impressive results. For acquisition size 10, Figure 7.6(a) and (b) show that the AL-kmeans strategy outperformed both the AL-entropy and AL-random strategies until approximately 100 data points were labeled. After that point, all query strategies performed more or less equivalent to each other.

Similar to the results observed with an acquisition size of 10, Figure 7.7(a) and (b) display that even for acquisition size 25, the AL-kmeans demonstrates a tendency for improved performance in the initial stages of the training process. The availability of a larger initial dataset diminishes the advantage of the AL-kmeans approach, as evidenced in Figure 7.7(c) and (d). In the case of the highest number of initial data points, Figure 7.7(d) demonstrates that the AL-entropy query strategy marginally surpasses the AL-random and AL-kmeans strategies in performance, starting at approximately 400 labeled data points, while AL-kmeans and AL-random exhibit similar performance.

For acquisition size 50, the AL-kmeans strategy shows a performance slightly more advantageous than other query strategies up to 250 data points. However, as shown in Figure 7.8(a) and (b), the difference is subtle. Independent of the initial training data quantity, the AL-random and AL-kmeans strategies tend to converge toward each other, resulting in equivalent F-scores. Notably, similar to acquisition size 10, the AL-entropy strategy demonstrates a slightly higher F-score upon the model's training completion for all levels of initially labeled data.

This might suggest that employing a diversity-based method is most effective when only a limited amount of labeled data is available. Presumably, this approach succeeded in selecting data points that more accurately embodied the dataset compared to the uncertainty-based method. As the amount of training data increased, the importance of selecting diverse data points seemed to diminish. As a result, AL-random and AL-kmeans displayed similar behavior, whereas AL-entropy achieved a marginally higher F-score. Nonetheless, the difference can be considered minimal, and the similar results are likely influenced by the inherent simplicity of the problem, as all query strategies exhibit high performance.

Contrary to the initial assumption that CEAL would effectively increase the amount of training data and subsequently enhance the model's performance, this does not appear to be the case. In instances where 0.5% of all data were labeled before training, Figures 7.6(c), 7.7(c), and 7.8(c) display that the inclusion of pseudolabeled data points only had a marginal effect on the model's performance and achieved an F-score comparable to AL-entropy alone. This can be attributed to the fact that only a few pseudolabeled samples exhibited entropy below the threshold and were subsequently added to the training set. This is not surprising since the presented results for the CEAL approach combine AL-entropy+CEAL-entropy. Instead, when the model is provided with more initially labeled data, as can be seen in Figures 7.6(d), 7.7(d), and 7.8(d), it exhibits increased confidence in its predictions, leading to a greater number of data points falling below the threshold and being incorporated into the training set. This results in inferior performance compared to other strategies, as the model is not sufficiently confident in its predictions, causing data points to be assigned incorrect labels. In light of these findings, the optimality of setting an initial threshold and subsequently reducing it by a factor over the number of epochs according to Equation 7.1 can be questioned. However, the poor results from the CEAL approach might also be because of the small amount of data used for training. If there was more training data, the estimations might be better and more certain, resulting in a better output from CEAL. That is, CEAL might be a good choice if data is less scarce, but in this work the focus is on a scenario where data annotation can be expensive, and it is therefore of interest to limit the annotation cost. Based on the results, the threshold-setting method appears to be more sensitive than has been proposed in previous studies [37]. An alternative approach, in which the threshold is more flexible and adapted to the model's confidence, might have yielded a different outcome. Another possible way could involve training the model over a greater number of epochs, thereby increasing the likelihood of accurate label classification, while simultaneously allowing the threshold to be set at a lower value.

Upon examining Table 7.2 to analyze the impact of acquisition sizes, it becomes apparent that the choice of acquisition size can influence the performance of different query strategies. Uncertainty-based query strategies, such as AL-entropy and CEAL-entropy, achieved a higher F-score from a smaller acquisition size over a greater number of epochs. For example, after 296 labeled data points, AL-entropy achieved an F-score of 0.89 with an acquisition size of 10, and 0.76 with an acquisition size of 50. The smaller acquisition size also enhanced the performance of both AL-random and AL-kmeans strategies up to 246 labeled data points. When further increasing the amount of labeled data, the difference can be seen as negligible. Upon the model reaching a meaningful performance level, the impact of acquisition sizes on the convergence rate dropped to a barely noticeable level. However, uncertainty-based query strategies, such as AL-entropy and CEAL-entropy, seem to benefit from a smaller acquisition size over an extended number of epochs.

To address the research question, a trade-off between time and F-score must be made. AL-kmeans utilized the [CLS] token, which had a size equal to the number of data points × the number of hidden states, to select data points for labeling by the oracle. Consequently, AL-kmeans might not be an appropriate strategy when working with high-dimensional data, if time consumption is a performance requirement. In contrast, AL-entropy and AL-random selected their data points based on probabilities for each label and, therefore, did not necessitate selection of data points from this high-dimensional space.

7.7.1 Limitations

It can be argued that labeling data needs to include a nonbiased oracle. In this study, this concern has been mitigated, as the oracle is a computer software that simulates a human in providing correct labels. However, a broader perspective and a possible future scenario includes a human annotator as the oracle. In such scenarios, a malicious oracle may introduce bias, for example, by consistently mislabeling tweets referencing a particular threat as negative [44].

The overall performance of the various AL approaches and query strategies was notably high, with several strategies achieving an F-score exceeding 0.90. This raises the question of whether the classification task itself is relatively straightforward for a complex transformer such as DistilBERT. Moreover, this level of performance is expected, given the binary nature of the classification problem, compared to a multiclass problem. Another consideration is that the model potentially learned the precise names of the 70 distinct APTs, which might have limited its ability to generalize and maintain comparable performance if new data containing different APTs are introduced.

In this work, a relatively small amount of data is used to train an LLM model. The experiments are simulations using an already-annotated dataset, aiming to replicate a scenario with a human expert annotating the data samples gradually, as they are selected over time by the AL strategy. For AL in general, however, it is important to also take the annotation cost into account, and arguably even more so when expert knowledge is needed for the annotation process. With humans, the task becomes more complex, possibly introducing a varying labeling cost, different types of noise, disagreement about the labels, etc. [11, 12]. While simulations have the advantage of providing more control over the experiments, they also risk oversimplifying the real-world scenario that is intended to be replicated.

7.8 CONCLUSIONS

This work investigated the potential of AL and its effectiveness for continuous improvement of classification of APTs in tweets. The transformer model DistilBERT was employed to classify the tweets, and AL approaches were utilized to iteratively add new labeled data points to the training dataset. Different AL approaches, including uncertainty-based and diversity-based query strategies, were examined, with several strategies achieving high performance. The diversity-based query strategy K-means excelled in the early training stages with limited prelabeled data. However, as the volume of training data increased, the performance advantage diminished. Additionally, as the number of training epochs increased, the uncertainty-based strategy showed a marginally improved performance relative to the other strategies. Interestingly, the CEAL approach did not enhance the model’s performance. The incorporation of data points with predicted labels often resulted in incorrect labels, thereby undermining the performance.

For future work, it would be interesting to explore the impact of combining the K-means strategy, which in this project demonstrated effectiveness when a minimal amount of labeled data was available, with uncertainty-based methods, such as entropy. This could be done by employing K-means to select *K* clusters and calculating entropy within each cluster, rather than on all data points in the unlabeled pool, which could potentially offer a more effective strategy for diverse sampling, while focusing on data points with higher model uncertainty. Additionally, assessing the generalizability of these findings across various datasets and distributions would be valuable. Moreover, experiments with human subjects in the annotation process would be of interest. This project focused on the performance of query strategies in a binary classification context, so extending the investigation to multiclass problems would be beneficial. Lastly, further examination of the potential of the CEAL approach is warranted, given its promising results in prior studies [37]. Exploring alternative methods for establishing the initial threshold, as well as for reducing the threshold, could prove beneficial.

ACKNOWLEDGMENTS

The authors would like to thank Katie Cohen and Leon Ericsson, Swedish Defence Research Agency, for comments on the manuscript.

NOTES

1. https://aclanthology.org/.
2. https://pypi.org/project/demoji/.

REFERENCES

1. U. Franke, A. Andreasson, H. Artman, J. Brynielsson, S. Varga, and N. Vilhelm, “Cyber situational awareness issues and challenges,” in *Cybersecurity and Cognitive Science*, A. A. Moustafa, Ed. London, United Kingdom: Academic Press, 2022, ch. 10, pp. 235–265, doi: 10.1016/B978-0-323-90570-1.00015-2.
2. B. Settles, *Active learning* (Synthesis Lectures on Artificial Intelligence and Machine Learning #18). Cham, Switzerland: Springer, 2012, doi: 10.1007/978-3-031-01560-1.
3. P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, “A survey of deep active learning,” *ACM Computing Surveys*, vol. 54, no. 9, pp. 1–40, 2021, Art. no. 180, doi: 10.1145/3472291.
4. A. Carp, J. Brynielsson, and A. Tegen, “Active learning for improvement of classification of cyberthreat actors in text fragments,” in *Proceedings of the 2023 22nd IEEE International Conference on Machine Learning and Applications (ICMLA 2023)*. Piscataway, NJ: IEEE, 2023, pp. 1279–1286, doi: 10.1109/ICMLA58977.2023.00193.

5. A. J. Joshi, F. Porikli, and N. Papanikolopoulos, “Multi-class active learning for image classification,” in *Proceedings of the 2009 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009).* Piscataway, NJ: IEEE, 2009, pp. 2372–2379, doi: 10.1109/CVPR.2009.5206627.
6. D. Tuia, M. Volpi, L. Copa, M. Kanevski, and J. Muñoz-Marí, “A survey of active learning algorithms for supervised remote sensing image classification,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 3, pp. 606–617, 2011, doi: 10.1109/JSTSP.2011.2139193.
7. H. M. S. Hossain, M. A. A. H. Khan, and N. Roy, “Active learning enabled activity recognition,” *Pervasive and Mobile Computing*, vol. 38, part 2, pp. 312–330, 2017, doi: 10.1016/j.pmcj.2016.08.017.
8. A. Tegen, P. Davidsson, and J. A. Persson, “Activity recognition through interactive machine learning in a dynamic sensor setting,” *Personal and Ubiquitous Computing*, vol. 28, no. 1, pp. 273–286, 2024, doi: 10.1007/s00779-020-01414-2.
9. F. Olsson, “A literature survey of active machine learning in the context of natural language processing,” Swedish Institute of Computer Science, Kista, Sweden, SICS Tech. Rep. T2009:06, 2009. [Online]. Available: https://urn.kb.se/resolve?urn=urn:nbn:se:ri:diva-23510
10. Y. Fu, X. Zhu, and B. Li, “A survey on instance selection for active learning,” *Knowledge and Information Systems*, vol. 35, no. 2, pp. 249–283, 2013, doi: 10.1007/s10115-012-0507-8.
11. A. Tegen, P. Davidsson, and J. A. Persson, “The effects of reluctant and fallible users in interactive online machine learning,” in *Proceedings of the IAL@ECML PKDD 2020 Workshop on Interactive Adaptive Learning. CEUR Workshop Proceedings*, 2020, pp. 55–71. [Online]. Available: https://ceur-ws.org/Vol-2660/ialatecml_paper4.pdf
12. A. Tegen, P. Davidsson, and J. A. Persson, “Active learning and machine teaching for online learning: A study of attention and labelling cost,” in *Proceedings of the 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA 2021).* Piscataway, NJ: IEEE, 2021, pp. 1215–1220, doi: 10.1109/ICMLA52953.2021.00197.
13. A. Tegen, P. Davidsson, and J. A. Persson, “A taxonomy of interactive online machine learning strategies,” in *Proceedings of the 2020 European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2020)*, vol. 2. Cham, Switzerland: Springer, 2020, pp. 137–153, doi: 10.1007/978-3-030-67661-2_9.
14. B. Settles, “From theories to queries: Active learning in practice,” in *Proceedings of the AISTATS 2010 Active Learning and Experimental Design Workshop.* JMLR Workshop and Conference Proceedings, 2011, pp. 1–18. [Online]. Available: https://proceedings.mlr.press/v16/settles11a.html
15. W. Newhouse, S. Keith, B. Scribner, and G. Witte, “National initiative for cybersecurity education (NICE) cybersecurity workforce framework,” National Institute of Standards and Technology, U.S. Department of Commerce, NIST Special Publication 800-181, 2017, doi: 10.6028/NIST.SP.800-181.
16. A. Sapienza, A. Bessi, S. Damodaran, P. Shakarian, K. Lerman, and E. Ferrara, “Early warnings of cyber threats in online discussions,” in *Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW 2017).* Piscataway, NJ: IEEE, 2017, pp. 667–674, doi: 10.1109/ICDMW.2017.94.
17. P. Chen, L. Desmet, and C. Huygens, “A study on advanced persistent threats,” in *Proceedings of the 15th IFIP TC 6/TC 11 International Conference on Communications and Multimedia Security (CMS 2014).* Berlin/Heidelberg, Germany: Springer, 2014, pp. 63–72.
18. Joint Task Force Transformation Initiative, “Managing information security risk: Organization, mission, and information system view,” National Institute of Standards and Technology, U.S. Department of Commerce, NIST Special Publication 800-39, 2011, doi: 10.6028/NIST.SP.800-39.

19. A. Lemay, J. Calvet, F. Menet, and J. M. Fernandez, "Survey of publicly available reports on advanced persistent threat actors," *Computers & Security*, vol. 72, pp. 26–59, 2018, doi: 10.1016/j.cose.2017.08.005.
20. T. Mattern, J. Felker, R. Borum, and G. Bamford, "Operational levels of cyber intelligence," *International Journal of Intelligence and CounterIntelligence*, vol. 27, no. 4, pp. 702–719, 2014, doi: 10.1080/08850607.2014.924811.
21. B. Miller, F. Linder, and W. R. Mebane Jr., "Active learning approaches for labeling text: Review and assessment of the performance of active learning approaches," *Political Analysis*, vol. 28, no. 4, pp. 532–551, 2020, doi: 10.1017/pan.2020.4.
22. Z. J. Wang, D. Choi, S. Xu, and D. Yang, "Putting humans in the natural language processing loop: A survey," in *Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing (HCINLP 2021)*. Stroudsburg, PA: Association for Computational Linguistics, 2021, pp. 47–52. [Online]. Available: https://aclanthology.org/2021.hcinlp-1.8
23. Z. Zhang, E. Strubell, and E. Hovy, "A survey of active learning for natural language processing," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP 2022)*. Kerrville, TX: Association for Computational Linguistics, 2022, pp. 6166–6190, doi: 10.18653/v1/2022.emnlp-main.414.
24. N. Stiennon, L. Ouyang, J. Wu, D. M. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano, "Learning to summarize from human feedback," in *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*. San Diego, CA: NeurIPS, 2020, pp. 3008–3021.
25. L. Zhang, J. Wu, D. Zhou, and G. Xu, "STAR: Constraint LoRA with dynamic active learning for data-efficient fine-tuning of large language models," 2024, arXiv: in *Findings of the Association for Computational Linguistics: ACL 2024*. Kerrville, TX: Association for Computational Linguistics, 2024, pp. 3519–3532, doi: 10.18653/v1/2024.findings-acl.209
26. Q. Hu, Y. Guo, X. Xie, M. Cordy, L. Ma, M. Papadakis, and Y. Le Traon, "Active code learning: Benchmarking sample-efficient training of code models," *IEEE Transactions on Software Engineering*, vol. 50, no. 5, pp. 1080–1095, 2024, doi: 10.1109/TSE.2024.3376964.
27. S. D. Bhattacharjee, A. Talukder, E. Al-Shaer, and P. Doshi, "Prioritized active learning for malicious URL detection using weighted text-based features," in *Proceedings of the 2017 IEEE International Conference on Intelligence and Security Informatics (ISI 2017)*. Piscataway, NJ: IEEE, 2017, pp. 107–112, doi: 10.1109/ISI.2017.8004883.
28. J. Lin, R. Luley, and K. Xiong, "Active learning under malicious mislabeling and poisoning attacks," in *Proceedings of the 2021 IEEE Global Communications Conference (GLOBECOM 2021)*. Piscataway, NJ: IEEE, 2021, pp. 1–6, doi: 10.1109/GLOBECOM46510.2021.9685101.
29. S. Moskal, and S. J. Yang, "Translating intrusion alerts to cyberattack stages using pseudo-active transfer learning (PATRL)," in *Proceedings of the 2021 IEEE Conference on Communications and Network Security (CNS 2021)*. Piscataway, NJ: IEEE, 2021, pp. 110–118, doi: 10.1109/CNS53000.2021.9705037.
30. S. Pal, Y. Gupta, A. Shukla, A. Kanade, S. Shevade, and V. Ganapathy, "ActiveThief: Model extraction using active learning and unannotated public data," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI 2020)*, vol. 34, no. 1. Palo Alto, CA: AAAI Press, 2020, pp. 865–872, doi: 10.1609/aaai.v34i01.5432.
31. T. Li, Y. Hu, A. Ju, and Z. Hu, "Adversarial active learning for named entity recognition in cybersecurity," *Computers, Materials & Continua*, vol. 66, no. 1, pp. 407–420, 2021, doi: 10.32604/cmc.2020.012023.
32. S. Srivastava, D. Gupta, B. Paul, and S. Sahoo, "A reinforced active learning sampling for cybersecurity NER data annotation," in *Proceedings of the 2022 OITS International Conference on Information Technology (OCIT 2022)*. Piscataway, NJ: IEEE, 2022, pp. 312–317, doi: 10.1109/OCIT56763.2022.00066.

33. B. Xie, G. Shen, C. Guo, and Y. Cui, “The named entity recognition of Chinese cybersecurity using an active learning strategy,” *Wireless Communications and Mobile Computing*, vol. 2021, no. 1, pp. 1–11, 2021, Art. no. 6629591, doi: 10.1155/2021/6629591.
34. C. E. Shannon, “A mathematical theory of communication,” *The Bell System Technical Journal*, vol. 27, no. 3, 4, pp. 379–423, 623–656, 1948, doi: 10.1002/j.1538-7305.1948.tb01338.x, 10.1002/j.1538-7305.1948.tb00917.x.
35. T. He, S. Zhang, J. Xin, P. Zhao, J. Wu, X. Xian, C. Li, and Z. Cui, “An active learning approach with uncertainty, representativeness, and diversity,” *The Scientific World Journal*, vol. 2014, pp. 1–6, 2014, Art. no. 827586, doi: 10.1155/2014/827586.
36. D. Arthur, and S. Vassilvitskii, “k-means++: The advantages of careful seeding,” in *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2007)*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.
37. K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, “Cost-effective active learning for deep image classification,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 12, pp. 2591–2600, 2017, doi: 10.1109/TCSVT.2016.2589879.
38. V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter,” 2019, arXiv: 1910.01108.
39. T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, “Transformers: State-of-the-art natural language processing,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP 2020)*. Stroudsburg, PA: Association for Computational Linguistics, 2020, pp. 38–45, doi: 10.18653/v1/2020.emnlp-demos.6
40. D. P. Kingma, and J. L. Ba, “Adam: A method for stochastic optimization,” in *Proceedings of the 2015 Third International Conference on Learning Representations (ICLR 2015)*, 2015, arXiv: 1412.6980.
41. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, vol. 1. Stroudsburg, PA: Association for Computational Linguistics, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.
42. H. Lilja, and L. Lundmark, “Tracking cyber threat actors in semi-automatic OSINT analysis,” in *Proceedings of the IST-190 Symposium on Artificial Intelligence, Machine Learning and Big Data for Hybrid Military Operations (AI4HMO)*. NATO Science and Technology Organization, 2021, pp. 1–12, Art. no. 31.
43. A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, “Bag of tricks for efficient text classification,” in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, vol. 2. Stroudsburg, PA: Association for Computational Linguistics, 2017, pp. 427–431. [Online]. Available: https://aclanthology.org/E17-2068/
44. B. Miller, A. Kantchelian, S. Afroz, R. Bachwani, E. G. Dauber, L. Huang, M. C. Tschantz, A. D. Joseph, and J. D. Tygar, “Adversarial active learning,” in *Proceedings of the 2014 ACM Workshop on Artificial Intelligent and Security Workshop (AISec 2014)*. New York, NY: ACM, 2014, pp. 3–14, doi: 10.1145/2666652.2666656.