Active Learning for Improvement of Classification of Cyberthreat Actors in Text Fragments

Amanda Carp*, Joel Brynielsson*†, Agnes Tegen†
*KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden
†FOI Swedish Defence Research Agency, SE-164 90 Stockholm, Sweden
Email: acarp@kth.se, joel@kth.se, agnes.tegen@foi.se

***Abstract*—In the domain of cybersecurity, machine learning can offer advanced threat detection. However, the volume of unlabeled data poses challenges for efficient data management. This study investigates the potential for active learning to reduce the effort required for manual data labeling. Through different query strategies, the most informative unlabeled data points were selected for labeling. The performance of different query strategies was assessed by testing a transformer model's ability to accurately distinguish tweets mentioning names of advanced persistent threats. The findings suggest that the K-means diversity-based query strategy outperformed both the uncertainty-based approach and the random data point selection, when the amount of labeled training data was limited. This study also evaluated the cost-effective active learning approach, which incorporates high-confidence data points into the training dataset. However, this was shown to be the least effective strategy.**

***Index Terms*—Active learning; natural language processing; cybersecurity; advanced persistent threat.**

I. Introduction

Machine learning (ML) holds great promise for detecting and responding to increasingly sophisticated cyberthreats. The large volume of available data can also present challenges for effective data management, however. Annotating data typically requires a significant amount of time and resources, particularly if the task necessitates specialized knowledge. 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 [1].

This work studies 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 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?

II. 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.

A. Active Learning

AL aims to reduce the labeling effort required to train a model [1]. 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 criteria. 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 [1]. Fig. 1 illustrates an iterative training process of active learning, which is commonly employed. The training

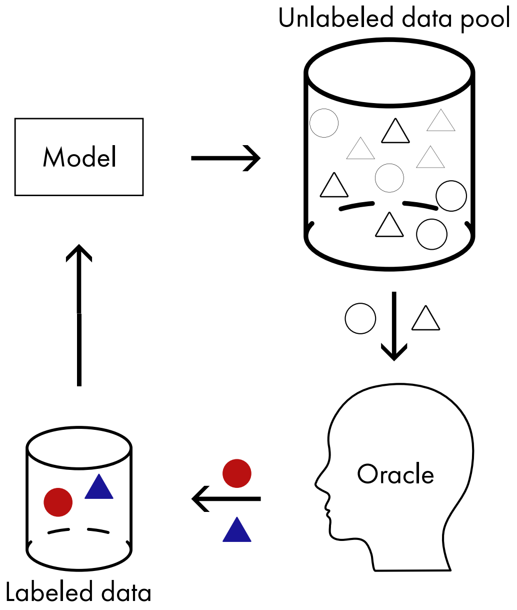


Fig. 1. Active learning cycle.

1946-0759/23/$31.00 ©2023 IEEE
DOI 10.1109/ICMLA58977.2023.00193

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 [2]. 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 [3] states that random sampling, which typically is used as a baseline, might be advisable if the problem is not well understood.

Several surveys explore AL within natural language processing (NLP) applications. Olsson [4] 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. [5] 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 [6] study, human-in-the-loop NLP frameworks are discussed from both the ML 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. [7] 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. [8] 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.

In this work, a relatively small amount of data is used to train an NLP 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.

B. The Cyberthreat

Traditional cyberthreat detection methods rely on preventive work and network monitoring [9]. 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 [10]. 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 [11]. 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 U.S. National Institute of Standards and Technology [12], 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 [13]. This can lead to serious confusion. APT28, for example, has multiple aliases, such as Fancy Bear, Strontium, Pawn Storm, Sofacy, Sednit, and Tsar Team [13]. 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 [9]. 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 [14]. At the tactical level, responses are often more time-sensitive. Associated necessary measures should be implemented more swiftly, 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 this project, 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.

[1] https://aclanthology.org/.

III. Active Learning Strategies

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 CEAL (CEAL-entropy).

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 [1]. 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 [15], is widely used in ML as a measure of uncertainty of a model.

Uncertainty sampling is prone to selecting outliers and data that may not accurately represent the dataset [16]. 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.

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 [17]. 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, we can infer 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 [17]. 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 Fig. 2. The unlabeled samples with an uncertainty measurement below a predetermined threshold δ are considered the most certain. The threshold for high-confidence sample selection is updated at each epoch, according to Equation 1. This is to be done to ensure that the labeling process remains dependable [17]. The threshold δ is defined by:

$$\delta = \begin{cases} \delta_0, & \text{for } t = 0, \\ \delta - dr \times t, & \text{for } t > 0. \end{cases} \quad (1)$$

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

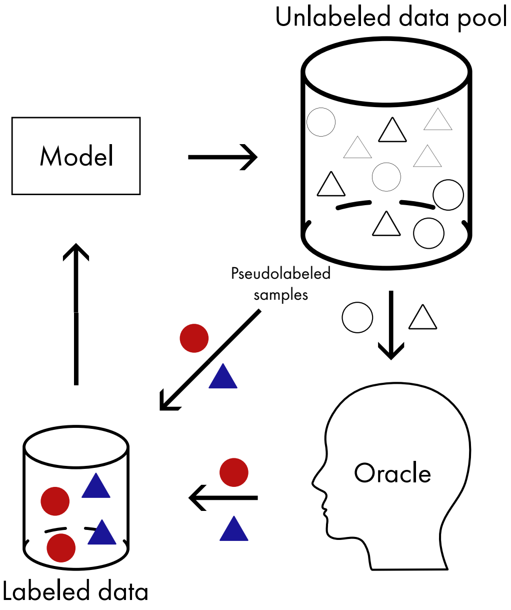


Fig. 2. Cost-effective active learning.

IV. Method

This section describes the model used, the data, and the experimental setup.

A. *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 [18]. By using a small model, the time and resource costs associated with model training can be reduced while still maintaining high performance. The pre-trained transformer DistilBERT, as described, was used through the Hugging Face library [19]. 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 non-padded ones. The stochastic optimizer Adam [20] 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 [21]. 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 pre-trained 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.

B. Dataset

The dataset used in the experiments contains approximately 35000 labeled tweets [22]. 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 vast 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 [23] 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 19000 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 percent negative samples. For clarification, this is presented in Table I.

TABLE I
Data distribution per class.

Dataset	Positive	Negative
Training	25%	75%
Validation	50%	50%
Unlabeled pool	34%	66%

C. Experimental Setup

Algorithm 1 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 ten 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. The pseudosamples, that is, samples chosen by CEAL, are also added.

Algorithm 1 Cost-effective active learning

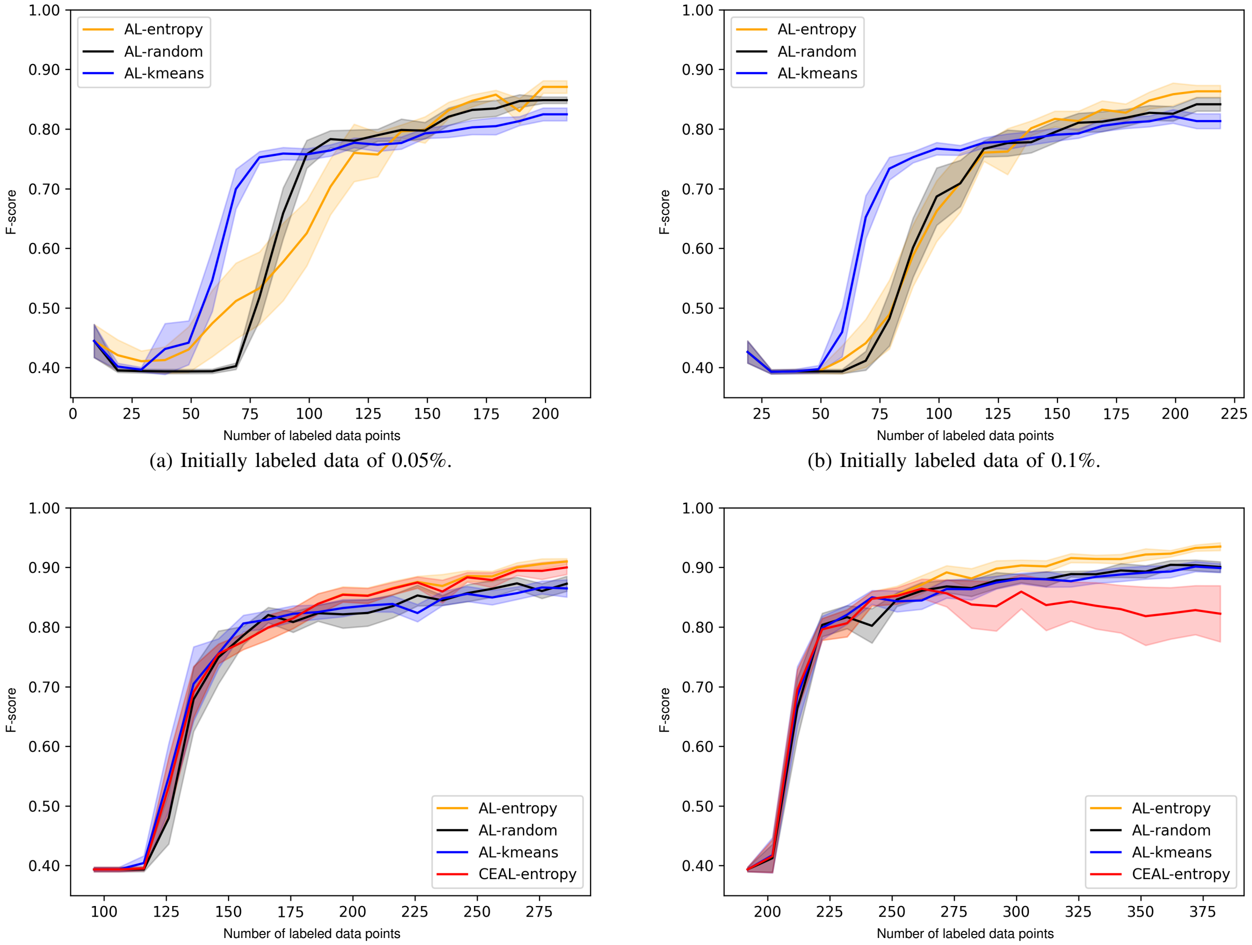
```
Input:
$\mathcal{M} \leftarrow$ Pre-trained transformer model,
$\mathcal{U} \leftarrow$ Unlabeled pool dataset,
$\mathcal{L} \leftarrow$ Initially labeled dataset,
$\mathcal{V} \leftarrow$ Validation dataset,
$\mathcal{K} \leftarrow$ Acquisition size for AL sampling,
$\delta \leftarrow$ Threshold for pseudosamples,
$N \leftarrow$ Maximum number of epochs.
Output: A trained model $\mathcal{M}$.
for $epoch = 0, 1, \ldots, N$ do
    if $epoch \neq 0$ then
        Put back pseudosamples from $\mathcal{L}$ to $\mathcal{U}$.
    end if
    Train model $\mathcal{M}$ with $\mathcal{L}$.
    Move $\mathcal{K}$ samples from $\mathcal{U}$ into $\mathcal{L}$ based on query strategies.
    Move $\mathcal{H}$ high-confidence pseudosamples from $\mathcal{U}$ into $\mathcal{L}$.
    Evaluate on $\mathcal{V}$ and log the results.
    Update $\delta$.
end for
```

The CEAL approach required optimal values for the threshold δ and the decay rate dr to be set in order to be implemented. The value δ set the limit for the number of samples that could be transferred to the labeled training dataset, and dr determined the rate of decay according to Equation 1. The decay rate was chosen to be 0.0033, as stated as the most optimal value according to the literature [17]. An initial threshold 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.

K-means clustering was used in an attempt 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, which corresponds to the last hidden state in the model. $\mathcal{K}$ data points were then chosen to be sent to an oracle for labeling, based on the smallest distance to each centroid.

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 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,

[2] https://pypi.org/project/demoji/.



(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%.

Fig. 3. Average F-score of AL approaches and query strategies with different amounts of initially labeled data and acquisition size 10, averaged across ten seeds and shown with 95% confidence intervals.

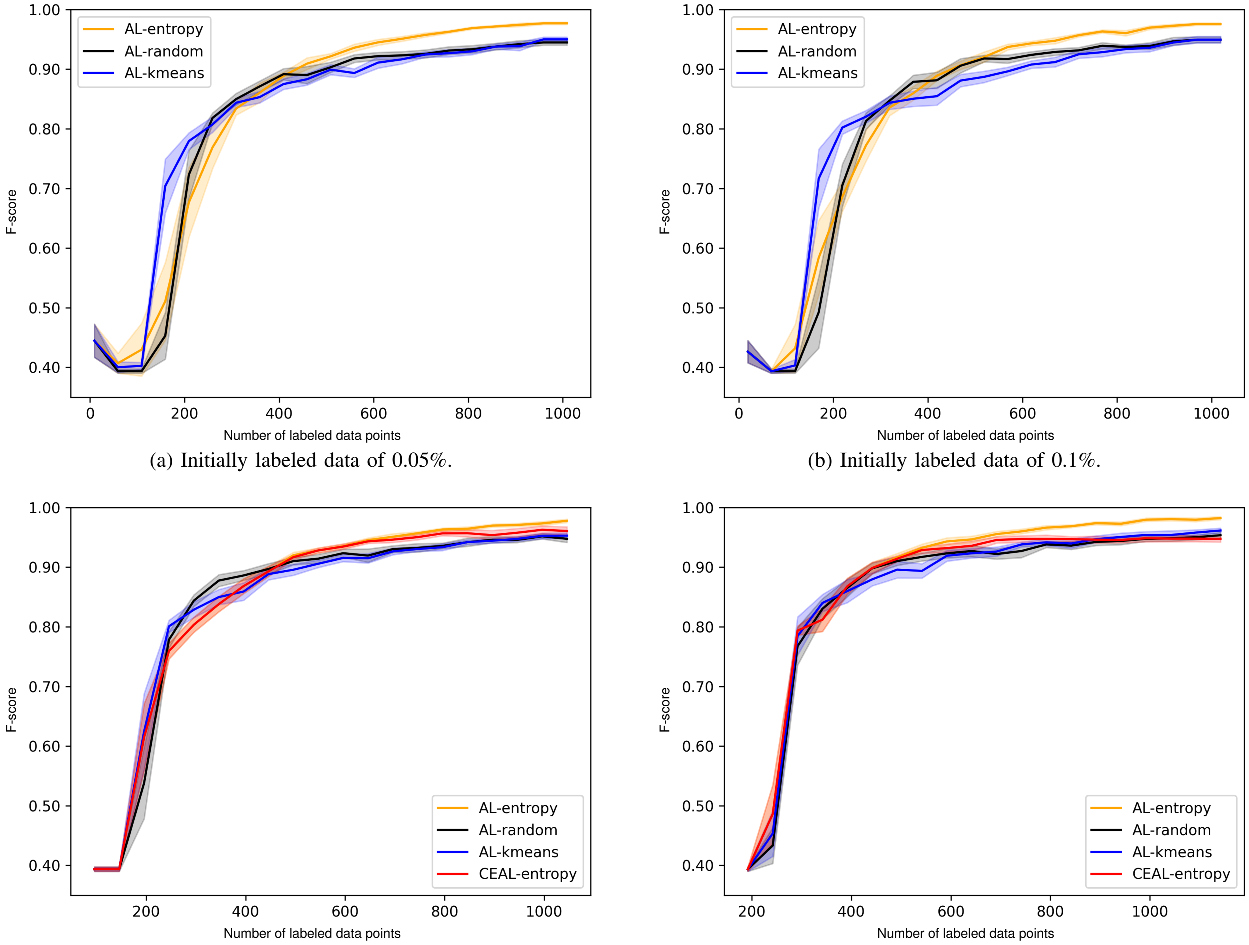
respectively. The acquisition sizes tested in the experiments were 10, 25, and 50 data points. The validation set was set to be 4% of the whole dataset. To prevent misleading results due to lack of data in the validation set, a consistent amount of data was allocated for validation, regardless of the 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.

V. Results

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 Fig. 3 and 4, the x-axis denotes the number of labeled data points by the oracle, not the pseudolabeled samples.

In Fig. 3, 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 amount 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.

In Fig. 4, 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 1000 data points (50×20) acquired from the pool of unlabeled data,



(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%.

Fig. 4. Average F-score of AL approaches and query strategies with different amounts of initially labeled data and acquisition size 50, averaged across ten seeds and shown with 95% confidence intervals.

to provide a comprehensive understanding of the performance trends under this experimental condition.

Table II 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 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.

TABLE II
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

VI. Discussion

As can be seen in Fig. 3 and 4, each of the tested query strategies yielded impressive results. For acquisition size 10, Fig. 3(a) and 3(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.

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 Fig. 4(a) and 4(b) the difference is subtle. Independent of the initial training data quantity, the AL-random and AL-kmeans strategies tend to converge towards 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, Fig. 3(c) and 4(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 Fig. 3(d) and 4(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, 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 [17]. 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 II 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 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.

A. *Limitations*

It can be argued that labeling data needs to include a non-biased 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.

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 multi-class 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.

Furthermore, the experiments were simulated, that is, human subjects were not used in the annotation process. With humans, the task becomes more complex, possibly introducing a varying labeling cost, noise or disagreement to the labels, etc. 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.

VII. 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 pre-labeled 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; especially, 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 multi-class problems would be beneficial. Lastly, further examination of the potential of the CEAL approach is warranted, given its promising results in prior studies [17]. Exploring alternative methods for establishing the initial thresholds, as well as reducing the thresholds, could prove beneficial.

REFERENCES

[1] 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.

[2] 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, 2021, pp. 137–153, doi: 10.1007/978-3-030-67661-2_9.

[3] 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

[4] 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

[5] 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.

[6] 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

[7] 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)*. Stroudsburg, PA: Association for Computational Linguistics, 2022, pp. 6166–6190, doi: 10.18653/v1/2022.emnlp-main.414.

[8] 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.

[9] 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.

[10] 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.

[11] 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, doi: 10.1007/978-3-662-44885-4_5.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," 2019, arXiv: 1910.01108.

[19] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, "HuggingFace's Transformers: State-of-the-art natural language processing," 2019, arXiv: 1910.03771.

[20] 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.

[21] 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.

[22] 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.

[23] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," 2016, arXiv: 1607.01759.