Appendix A: Sequence Diagram Overview

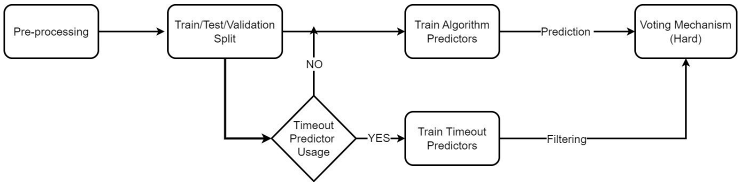

Figure 1: The diagram depicts the process of passive learning, which includes preprocessing, splitting the dataset, and training multiple binary Random Forest (RF) algorithm classifiers and timeout classifiers, followed by a hard voting mechanism to finalize predictions. Timeout predictors are used for filtering algorithm predictors in the voting mechanism. All models are included in the voting mechanism where the timeout predictor configuration is not applied.

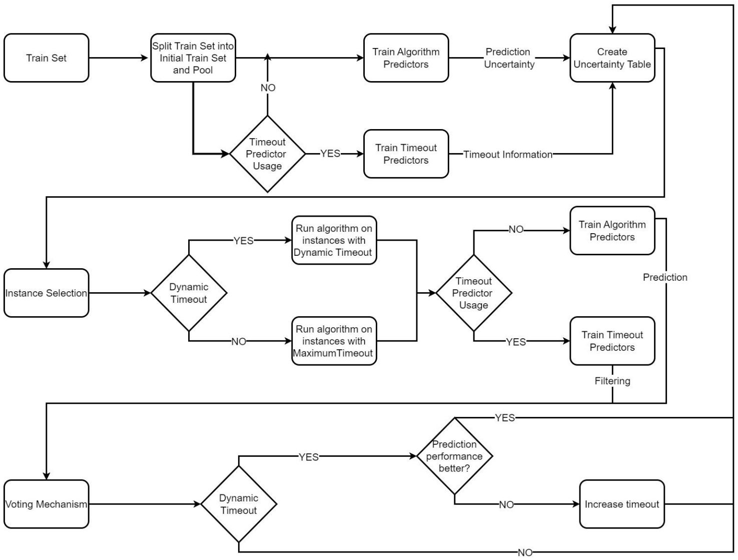

Figure 2: The diagram illustrates the steps in the proposed approach. It starts with splitting the training set into an initial training set and a pool for each model. Multiple binary Random Forest (RF) algorithm and timeout classifiers are then trained. The instance selection process involves identifying the most uncertain data points across all models and excluding instances predicted to time out by the timeout predictors. A dynamic timeout is applied during the labeling process, which is increased when there is no performance enhancement on the validation set. After labeling, the iterative process begins again, continuously refining the models.

Appendix B: Analysis of Uncertainty Measurement Behaviours in Active Learning for Binary Classification

There are three main approaches for uncertainty sampling in active learning. However, in a binary classification setting (which is what we use) these approaches perform identically to each other. We explain the different approaches here. Figure 3 shows the behaviour of these uncertainty sampling methods graphically.

We implement ‘Least Confidence‘ in our approach.

- *Least Confidence*: for a given input x and an output label $\hat{y}$, we can measure the posterior probability $P(\hat{y}|x;\theta)$ of observing $\hat{y}$ given x via the current model (parameterised by θ). The Least Confidence method selects data points x^* with the smallest maximum posterior probability across all labels:
$$x^* = \underset{x}{\operatorname{argmin}} \max_{\hat{y}} P(\hat{y}|x;\theta) \tag{1}$$

- *Margin-based*: this approach takes the two highest posterior probability values for each input data point x and calculates their difference. The smaller the difference, the less certain the model is about its prediction and vice versa. More formally, let $\hat{y_1}$ and $\hat{y_2}$ the output labels with the highest and second-highest posterior probabilities for a given input x, respectively, the queried points x^* are chosen as:
$$x^* = \underset{x}{\operatorname{argmin}}\, P(\hat{y_1}|x;\theta) - P(\hat{y_2}|x;\theta) \tag{2}$$

- *Entropy-based*: this approach takes into account the posterior probability values across *all* output classes. The idea is to select the data points x^* where there is a high entropy among the predicted output labels:
$$x^* = \underset{x}{\operatorname{argmax}} - \sum_{i} P(\hat{y}|x;\theta) \log P(\hat{y}|x;\theta) \tag{3}$$

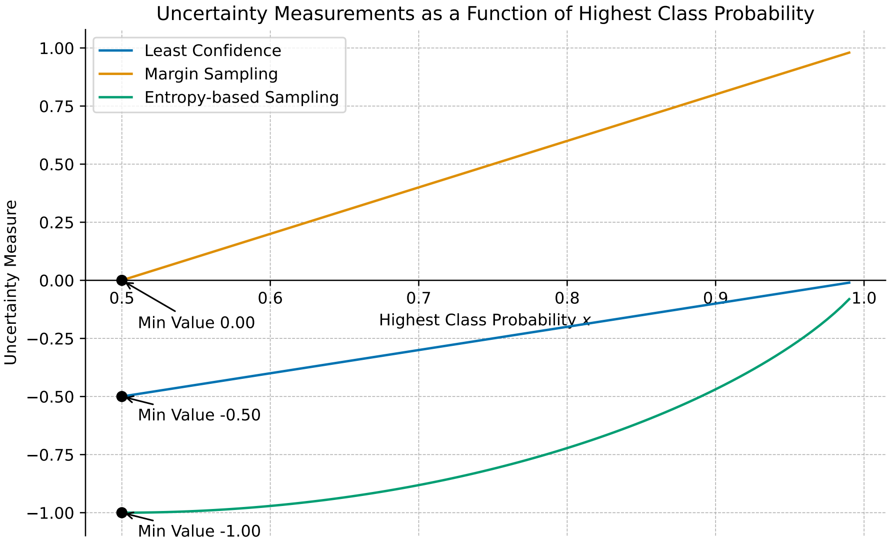

Figure 3: Uncertainty measurements as a function of the highest class probability. The red curve represents the Least Confidence uncertainty (LC) calculated as $LC = x - 1$, the green curve denotes Margin Sampling (MS) using the formula $MS = x - (1 - x)$, and the blue curve illustrates the Entropy-based method ($H(x) = -[x \log_2(x) + (1 - x) \log_2(1 - x)]$). Critical minimum values for each method are marked with black circles and annotated to emphasise the points where the uncertainty function is minimised.

Appendix C: Performance of 8 individual configurations

Figure 4 illustrates a side-by-side comparison of the following eight active learning strategies in binary classification without aggregation across configurations:

- Uncertainty Sampling without Timeout Predictor & without Dynamic Timeout (NO TO & NO DT)
- Uncertainty Sampling with Timeout Predictor (TO)
- Uncertainty Sampling with Dynamic Timeout (DT)
- Uncertainty Sampling with Timeout Predictor and Dynamic Timeout (TO+DT)
- Random Sampling without Timeout Predictor & without Dynamic Timeout (NO TO & NO DT)
- Random Sampling with Timeout Predictor (TO)

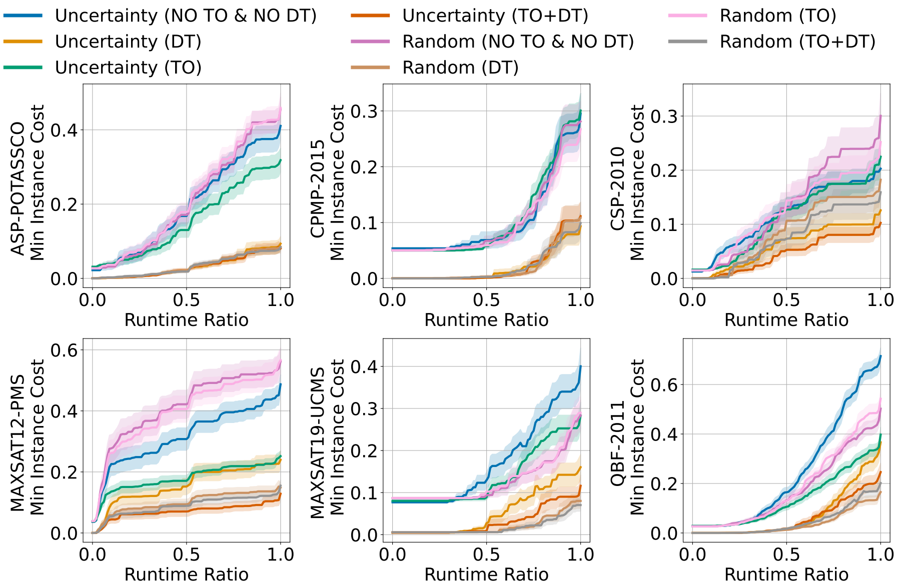

Figure 4: Comparison of performance across eight configurations as described in the paper. Each configuration was normalised according to the passive learning prediction performance ratio.

- Random Sampling with Dynamic Timeout (DT)
- Random Sampling with Timeout Predictor and Dynamic Timeout (TO+DT)

Appendix D: Experimental Setup

This study used a Random Forest classifier configured with 100 estimators and the Gini impurity measure to determine the best splits. Each tree is limited to using up to the square root of the number of features, and the depth of the decision trees is practically unlimited (with a maximum depth set to 2^{31}). Nodes require at least two samples before splitting, and bootstrapping is enabled for sampling data when building each decision tree. These settings were determined through experimentation in the passive learning setup and were consistently used throughout the study.

We also addressed missing data by removing features where more than 20% of the instances had missing values and applied a median imputer to fill the remaining gaps.

We employed a cross-validation approach with 10 splits to validate the robustness of our study. To ensure reproducibility, we used 5 distinct seeds (7, 42, 99, 123, 12345) across our experiments, ensuring consistent generalization across multiple runs.

To determine when to increase the timeout in configurations where dynamic timeout is used, 10% of the training set was allocated as the validation set. Throughout the experiments, timeout values were scaled by a factor of 10, following the PAR10 measure.

Additional Parameters and Configurations:

Timeout Predictor Usage: This parameter determines whether the timeout predictor is used on the system.

Timeout Limit: Sets the initial time for the dynamic timeout. We used an initial timeout of 100 seconds when employing dynamic timeout, and a fixed timeout of 3600 seconds when not using dynamic timeout.

Timeout Increase Rate: Adjusts the dynamic timeout when there is no improvement in prediction performance on the validation set. We set this rate to increase by 100 seconds when no improvement was observed.

Initial Train Size: Determines the size of the initial training set for uncertainty selection. The initial training set was created by randomly selecting 20 data points from the overall training set.

Query Size: Refers to the percentage of the dataset queried in each iteration. We set this to 1%, meaning 1% of the total pool of candidates was queried in each iteration of our experiments.

For active learning, we utilized the modAL framework [**?**], which facilitated the implementation of uncertainty sampling and other active learning strategies in our experiments.

Appendix E: Description Table of Selected Datasets

Table 1 shows key information about the datasets used in this study. It includes the time it took for the algorithms to run, the Virtual Best Solver(VBS) representing the best algorithm for each problem, and the Single Best Solver (SBS) as the best overall algorithm. While VBS is the hypothetical best, SBS serves as a benchmark for comparison against other algorithms.

Dataset	Instances	Algorithms	Features	Total Time	VBS	SBS
ASP-POTASSCO	1294	11	138	2,085h	8h	112h
CPMP-2015	527	4	22	682h	33h	134h
CSP-2010	2024	2	86	435h	49h	82h
MAXSAT12-PMS	876	6	37	1,472h	8h	85h
MAXSAT19-UCMS	572	7	54	545h	20h	52h
QBF-2011	1368	5	46	352h	28h	300h

Table 1: Descriptive statistics of selected datasets. Times rounded to the nearest whole number.

Appendix F: Timeout (TO) Configuration Impact on Passive Learning

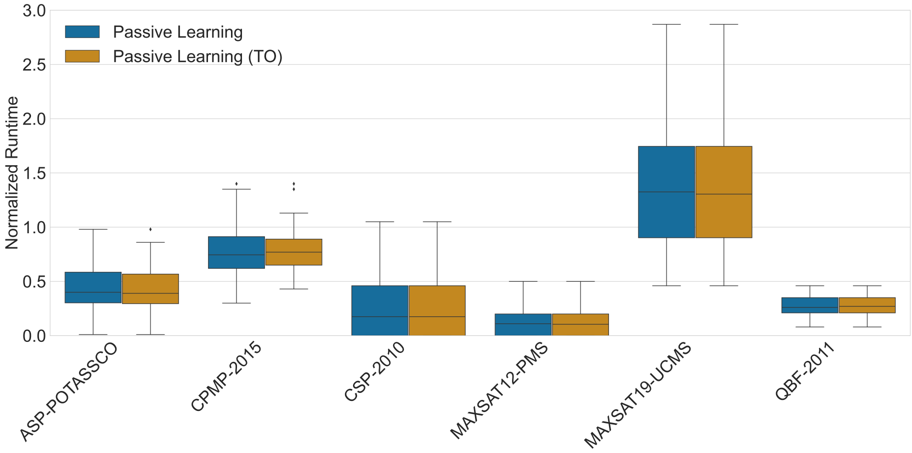

Figure 5: Comparison of Timeout (TO) Configuration Impact on Passive Learning: The graph illustrates that implementing the TO configuration in passive learning on the test set does not significantly enhance performance, yet importantly, it does not compromise prediction accuracy either.