

Online Hierarchical Partitioning of the Output Space in Extreme Multi-Label Data Streams

Lara Neves^{a,*}, Afonso Lourenço^{a,**}, Alberto Cano^b and Goretí Marreiros^a

^aGECAD, ISEP, Polytechnic of Porto, Rua Dr. António Bernardino de Almeida, Porto, 4249-015, Portugal

^bVirginia Tech, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061 USA

ORCID (Lara Neves): <https://orcid.org/0009-0000-3530-1848>, ORCID (Afonso Lourenço):

<https://orcid.org/0000-0002-3465-3419>, ORCID (Alberto Cano): <https://orcid.org/0000-0001-9027-298X>, ORCID

(Goretí Marreiros): <https://orcid.org/0000-0003-4417-8401>

Abstract. Mining data streams with multi-label outputs poses significant challenges due to evolving distributions, high-dimensional label spaces, sparse label occurrences, and complex label dependencies. Moreover, concept drift affects not only input distributions but also label correlations and imbalance ratios over time, complicating model adaptation. To address these challenges, structured learners are categorized into local and global methods. Local methods break down the task into simpler components, while global methods adapt the algorithm to the full output space, potentially yielding better predictions by exploiting label correlations. This work introduces iHOMER (Incremental Hierarchy Of Multi-label Classifiers), an online multi-label learning framework that incrementally partitions the label space into disjoint, correlated clusters without relying on predefined hierarchies. iHOMER leverages online divisive-agglomerative clustering based on *Jaccard* similarity and a global tree-based learner driven by a multivariate *Bernoulli* process to guide instance partitioning. To address non-stationarity, it integrates drift detection mechanisms at both global and local levels, enabling dynamic restructuring of label partitions and subtrees. Experiments across 23 real-world datasets show iHOMER outperforms 5 state-of-the-art global baselines, such as MLHAT, MLHT of Pruned Sets and iSOUPT, by 23%, and 12 local baselines, such as binary relevance transformations of kNN, EFDT, ARF, and ADWIN bagging/boosting ensembles, by 32%, establishing its robustness for online multi-label classification.

1 Introduction

Data stream mining aims to learn models from evolving data, where concept drift alters the underlying distribution over time [15]. Traditionally, each instance is assumed to have a unique label, but real-world objects often carry multiple, interdependent meanings. This gives rise to streaming multi-label learning (SMLL), where the goal is to predict sets of labels while accounting for label dependencies [32]. SMLL faces three major challenges: (1) exponential output space; (2) class imbalance, both within and across labels; and (3) concept drift, which dynamically changes label definitions and correlations [1]. These challenges are especially acute in streaming settings, where models must adapt continuously and efficiently.

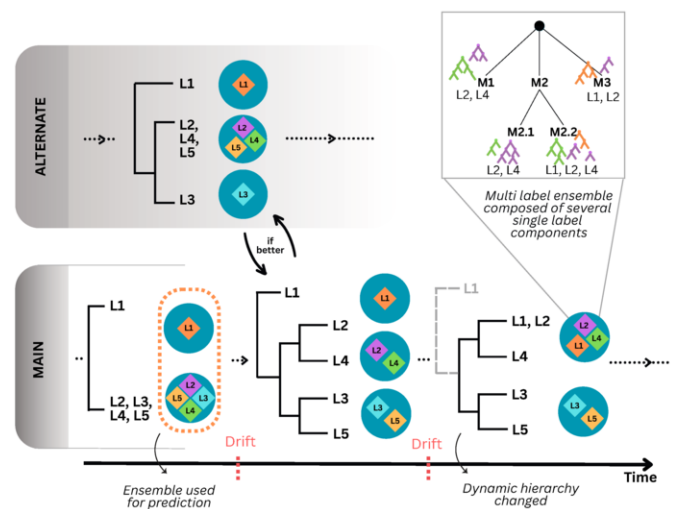


Figure 1: Hierarchy of label clustered classifiers evolves over time.

Local vs. Global Models. Structured prediction methods are typically divided into local and global approaches [21]. Local methods decompose multi-label tasks into independent scalar predictions, while global methods train a single model to output structured predictions directly [11, 29], often yielding superior performance by better capturing label dependencies. To combine the strengths of both, output specialization methods model label subsets with global predictors, using label space partitioning to create clusters that lie between fully local and global extremes [39]. HOMER [44] uses hierarchical decomposition to simplify learning in high-dimensional spaces, while more recent work like HPML [16] explores flat, non-hierarchical clusters to balance accuracy and efficiency.

This work. Building on these ideas, this work proposes iHOMER (Incremental Hierarchy Of Multi-label classifiers), an online approach that clusters the label space into disjoint, correlated groups, forming hybrid partitions that combine global and local strategies (Figure 1). iHOMER does not rely on fixed hierarchies and avoids expensive tree construction while preserving partition coherence. Experiments on 23 real-world datasets show iHOMER outperforms 5 global baselines by 23%, 12 local baselines by 32%, and random search by 40%, validating its effectiveness. Source code, datasets, and results are publicly available ².

* Corresponding Author. Email: lpspn@isep.ipp.pt.

** Corresponding Author. Email: fonso@isep.ipp.pt.

¹ Equal contribution.

2 Related work

Multi-label learning algorithms model label correlations in various ways, which can be categorized along four dimensions: (1) instance location (global vs. local), (2) dependency (conditional vs. unconditional), (3) dataset space (feature, instance, or label), and (4) correlation order (first-, second-, or high-order) [46]. Global correlations consider the full dataset, while local ones work within subsets like clusters. Conditional correlations depend on instance features, whereas unconditional ones model co-occurrence across all data. Correlations may arise from the feature, instance, or label space. First-order models ignore label dependencies, e.g., Binary Relevance (BR), second-order models capture pairwise relations, and high-order methods model interactions among multiple labels. Despite these differences, these multi-label strategies share patterns: global/high-order/unconditional methods model the full label set, while local/second-order/conditional methods target subsets. More interestingly, these can be grouped into three algorithmic families: local models, global models, and hybrid partitions.

Local approaches. Problem transformation methods fit the data for traditional single-label streaming classifiers. In particular, BR is commonly used for its scalability and parallelism [7], though it suffers from class imbalance and empty predictions. Streaming BR solutions include label-wise weighting [31], separate windows for positives and negatives [40], and concept drift detection via label entropy [43]. MINAS-BR handles novelty and unlabeled labels in extreme streaming settings [20]. Beyond BR, second-order methods like Label Ranking use one-vs-one classifiers to model pairwise label orderings, though scalability is an issue in high dimensions [23]. High-order methods like Label Combination treat label sets as multi-class targets [38], while managing the growing number of label sets with probabilistic modeling [45] or early-sample buffering [32].

Global approaches. Algorithm adaptation methods directly handle multi-label outputs, reducing complexity and making them ideal for streaming. Notable examples include extensions of ML-kNN [46], using Bayesian inference over neighbors, and extensions of ML-DT [9], using multi-label entropy. Streaming ML-kNN solutions include windowing to efficiently update posteriors [40] and dual-memory models like ML-SAM-kNN [34], which splits recent and long-term data. Extensions such as MLSAMPkNN [35] and ML-SAKNN [36] introduce punitive mechanisms and adaptive clustering. ARkNN [37] proposes resource-efficient instance aging. Streaming ML-DT solutions extend Hoeffding Trees (HT) [10] with label-set predictors and pruning techniques. For example, iSOUP [29] applies multi-target regression via perceptron-based model trees, which can be extended with drift detection [41] and per-target model selection [25]. MLHAT [13] goes beyond, incorporating drift and label imbalance adaptation via cardinality-aware metrics.

Hybrid partitions. It is important to note that the aforementioned global approaches are sometimes referred to as ensemble methods due to their use of multiple binary models. Similarly, hybrid partitioning methods are occasionally labeled as ensemble methods because they involve several global multi-label models. However, since these models handle disjoint label subsets and collectively represent a unified prediction system, they effectively constitute a single multi-label model. Hybrid partition methods blend local and global models by clustering labels and applying global predictors to each subset [39], offering a middle ground between fine-grained and holistic learning. By modeling label correlations locally, i.e., within label clusters [19], or across all labels [16], these methods aim to efficiently guide label space partitions, rather than relying on random

partitioning [42, 8] or genetic algorithms that overlook label dependencies [4, 28]. In fact, examples of this can be found beyond tree-based methods, leveraging the modular nature of rule learning systems to build localized models over partitions of the input space. For example, AMRules [2] has been extended to multi-label classification [12], incorporating mechanisms for handling concept drift using the Page-Hinkley test and detecting anomalous instances. This modularity enables surpassing both global and local correlation-based methods via rule specialization for non-disjoint outputs in multi-target regression [11] and multi-label classification [39]. When performing rule expansion, if the variance in the target space is reduced only for certain targets, a new rule is created for those targets, while a complementary rule is retained for the rest. As a result, the system can generate rules covering all, some, or just a single target.

iHOMER. Building on these advancements, iHOMER is the first incremental tree-based output specialization approach, whose main novelty comes from 4 distinct characteristics: (1) flexibility, (2) non-repetition, (3) computational efficiency, and (4) adaptive mechanisms. Firstly, while some algorithms require high parameterization for searching the best output space partition, such as defining the number of individuals, the number of generations, the number of classifiers, and the number of labels for each classifier in genetic algorithms [4, 28], iHOMER allows the creation of various hybrid partition configurations without the need to previously determine their number. Secondly, iHOMER ensures that labels are not repeated in label clusters for each hybrid partition generated by adjusting cluster labels to satisfy this constraint, which is not done in related methods [30, 42]. Thirdly, iHOMER leverages extremely resource-efficient algorithm choices, e.g., a growing clustering structure based on an incremental *Jaccard* dissimilarity measure, and a growing tree structure based on a multivariate *Bernoulli* process, both of which are statistically guided by the *Hoeffding bound* [10]. Fourthly, iHOMER incorporates drift detection mechanisms at both global and local levels, enabling dynamic restructuring of label partitions and subtrees, validated with an alternate background replacement process [5].

3 Methodology

iHOMER is an online approach that clusters the label space to create hybrid partitions. Firstly, correlations between labels are modeled at a dataset level with the *Jaccard* similarity measure. Secondly these are clustered in order to obtain hybrid partitions of the label space, with a top-down growing hierarchy of splits governed either by a user-defined threshold on the number of observations, denoted n_{\min} , or by a statistically grounded criterion based on the *Hoeffding bound*. Thirdly, as the data distribution evolves, rendering earlier partitioning decisions obsolete, previous splits and child nodes are re-aggregated into their parent node, also based on the *Hoeffding bound*. Moreover, since iHOMER should extract the most meaningful and balanced clusters from the hierarchy at any given time, a second-stage balanced aggregation is performed when necessary, according to a minimum safe cluster size determined via a majority Labelset classifier analyzing the distribution of labelset sizes. Subsequently, a drift detection mechanism monitors the multi-label accuracy, allowing it to trigger an alternate clustering process in the background, which replaces the main one if the concept drift is confirmed. Finally, a global multi-label incremental tree-based learner is applied for each constructed partition. It is important to note that since this ensemble of global models is concerned with disjoint sets of labels, iHOMER in fact constitutes a single multi-label model.

²<http://github.com/larasneves/iHomer>

3.1 Incremental dissimilarity measure

To effectively capture pairwise similarity degrees of all labels in the label space, the *Jaccard* index was used [16]. Formally, considering p_j and q_j as two distinct labels within the label space, the label dissimilarity is defined as:

$$d(\text{Jaccard}) = \left(1 - \frac{a}{a + b + c}\right) \quad (1)$$

where a represents the number of instances simultaneously classified with both p_j and q_j ; b denotes the number of instances classified only with p_j ; and c is the number of instances classified only with q_j . It is important to note that the factors used to compute the *Jaccard* index can be updated incrementally. As each example is processed once, sufficient statistics are updated incrementally, while the dissimilarity graph, shown in Figure 2, is only computed when testing for splitting or aggregation. Missing labels are imputed as zero.

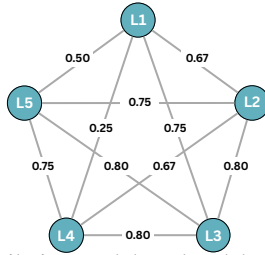


Figure 2: Dissimilarity graph based on label co-occurrences.

3.2 Growing output space hierarchy

The iHOMER algorithm employs online divisive agglomerative clustering to incrementally construct a tree-structured hierarchy that reflects the nested organization of label clusters. This process follows a top-down strategy, as illustrated in Figure 3. The leaves of the tree correspond to the resulting clusters, each associated with a distinct subset of variables [33]. The union of all leaves comprises the complete variable set, and the intersection between any two leaves is empty. During the growing process, based on the incremental dissimilarity measure and the corresponding diameters of the clusters, a key decision is made on whether a given leaf node, i.e., cluster, should be split. This cluster splitting is governed either by a user-defined threshold on the number of observations, denoted n_{\min} , or by a statistically grounded *Hoeffding bound* criterion, which provides a concentration inequality for the mean of a bounded random variable. Specifically, for a real-valued random variable r bounded in range R , after observing n independent samples, the bound ensures that, with confidence $1 - \delta_{cluster}$, the true mean lies within ϵ of the empirical mean, where:

$$\epsilon = \sqrt{\frac{R^2 \ln(1/\delta_{cluster})}{2n}}. \quad (2)$$

Each cluster C_k maintains its own Hoeffding threshold ϵ_k to determine whether the diameter-based heuristic condition is met. In particular, the splitting criterion compares the spread of pairwise dissimilarities within the cluster to the deviation from the mean:

$$\frac{(d_1 - d_0)}{|d_1 + d_0 - 2d|} > \epsilon_k. \quad (3)$$

where d_1 is the maximum distance between any two elements in the cluster, d_0 is the minimum pairwise distance, and d is the average dissimilarity across all pairs. If this inequality holds, the cluster is considered sufficiently diverse to justify splitting. In this case, the

observations corresponding to the pair with maximum dissimilarity are selected to initiate new clusters as pivot variables, and the remaining variables are assigned to the nearest new cluster based on pivot proximity. This results in two subclusters centered around the most dissimilar elements in the original cluster. While this process enables the iHOMER tree to grow in a data-driven and statistically sound manner, maintaining robustness against noise and avoiding over-segmentation, near-equidistant variables can delay the decision between two equally valuable splits. To address this, a parameter $\tau_{cluster}$ is introduced. If $\tau_{cluster} > \epsilon_k$, the system forces a test, assuming enough examples have been seen.

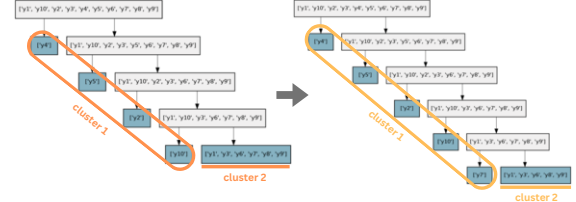


Figure 3: Dynamic hierarchical label clusters in Hypersphere dataset.

3.3 Aggregating output space hierarchy

A critical feature of iHOMER lies in its continuous monitoring of cluster diameters, which provides a basis not only for initiating splits but also for detecting when prior splits may no longer reflect the current structure of the data. In stationary data streams, each successive split within a divisive hierarchical clustering scheme typically leads to a reduction in intra-cluster dissimilarity, producing increasingly refined subgroups. However, in non-stationary environments, the data distribution may evolve, rendering earlier partitioning decisions obsolete. In such cases, the system must possess the flexibility to reverse previous splits and re-aggregate child nodes into their parent. To detect when a previous split has become inadequate, the algorithm assesses whether the diameters of sibling clusters remain meaningfully smaller than that of their parent:

$$2 \cdot \text{diam}(C_j) - (\text{diam}(C_k) + \text{diam}(C_s)) < \epsilon_j. \quad (4)$$

where C_k is a given leaf node, C_s its sibling, C_j their common parent, and ϵ_j the Hoeffding-based confidence interval derived from the amount of data previously accumulated at the parent node. If this condition holds, then there is insufficient evidence to justify maintaining the split, and the system chooses to re-aggregate C_k and C_s into their parent C_j . The resulting re-aggregated node begins with fresh statistical summaries, under the assumption that only future observations will inform its internal structure. This initialization allows the system to remain responsive to evolving data patterns. Moreover, the dual use of the *Hoeffding bound*, for both splitting and re-aggregation, enhances iHOMER's ability to adapt to changing data distributions while maintaining a balance between sensitivity to novel patterns and stability against noise. However, the resulting clusters might be unbalanced. Ideally, iHOMER should extract the most meaningful and balanced clusters from the hierarchy at any given time, as illustrated in Figure 3. To maintain flexibility of the hierarchy building process, a second-stage aggregation is performed only when necessary, ensuring clusters represent coherent groups rather than isolated individual labels. Leveraging on the fact that the produced dendrogram can be cut at different levels, this process adjusts the tree structure by merging clusters when appropriate, refining the representation of the data over time. To achieve this, the Majority Labelset classifier is used, when requested, to analyze the distribution of labelset sizes. It retrieves a frequency table indicating

how often labelsets with a given number of labels occur, using this information to determine the minimum safe cluster size. This approach is intuitive as it suggests that clusters should contain at least as many labels as the most common labelset size. For example, if the most frequent labelset size in a dataset is eight, predicting only two labels at a time would be inefficient. Conversely, if most instances have just one label, maintaining specialized models for single-label prediction is more effective. Since this strategy follows a hierarchical clustering approach, imposing a maximum number of labels per cluster is unnecessary and may hinder the natural refinement of the structure over time. Theoretically, all labels could initially form a single cluster, but as new observations are processed, the hierarchy self-adjusts, splitting or merging clusters as needed. This adaptive process ensures that the clustering structure evolves to balance granularity and coherence, making rigid constraints on cluster size unnecessary.

3.4 Alternate output space partitioning

As the data distribution evolves, rendering earlier partitioning decisions obsolete, iHOMER dynamically aggregates or expands its label partitioning, allowing faster recovery rates. However, the base learner performance still requires time to improve with the continuous arrival of new samples, as the historical data before the concept drift plays a restraining role in the model's update, leading to the inability of classification accuracy to recover in a short time. In other words, if the learning model has been in a state of fail-safe operation, i.e., carrying the original distributed samples or information, for a long time, it might not be able to adapt to the demand of high timeliness in streaming data. Furthermore, in addition to true concept drifts, iHOMER is sensitive to false concept drifts, such as deviation of noise samples from the normal distribution, and unstable representations due to updated small-scale datasets. Consequently, to smooth these transitions, iHOMER features a conservative replacement strategy, by selectively retiring outdated models [5]. When iHOMER detects multiple consecutive drift signals, suggesting the active learner may be misaligned with the data stream, an alternative learner is initialized. This alternate component explores a new representation of the data stream and can replace the active model only if it demonstrates superior performance in terms of current subset accuracy. Specifically, the *Welch's T-test* is applied to assess whether the prediction errors $\alpha, \alpha_{c'}$ of the active label clustering strategy c and a candidate alternate c' differ significantly, under the assumption of unequal variances. The test statistic $t_{statistic}$, of confidence $1 - \delta_{alt-cluster}$, is computed as:

$$t_{statistic} = \frac{\alpha_c - \alpha_{c'}}{\sqrt{\frac{\sigma_c^2}{N_c} + \frac{\sigma_{c'}^2}{N_{c'}}}} \quad (5)$$

where ϵ denotes the mean error, σ^2 the error variance, and N the number of instances observed in an *ADWIN*-controlled sliding window for each model [5]. When the test indicates a statistically significant difference, the alternative partitioning may replace its counterpart, supporting dynamic adaptation to concept drift in the data stream. Moreover, recursive replacements are possible, with multiple competing alternatives, specializing in different temporal regions of the data stream. The more recent alternate learner typically adapts quickly to current trends, making it particularly effective when dealing with abrupt concept drifts. In contrast, the older and generally more stable active learner adapts more slowly but tends to perform better in stationary or gradually evolving segments of the stream.

This mechanism enables dynamic retirement of underperforming active models while preserving stability. Moreover, the introduction of a *grace period* n_{min} before replacement decisions helps reduce false alarms and stabilizes system responsiveness in the face of noise and short-term fluctuations.

3.5 Global base learner

Once iHOMER has identified meaningful label clusters through hierarchical partitioning, a specialized multi-label Hoeffding adaptive tree (MLHAT) is trained on each cluster. Following the principles of incremental Hoeffding trees [10], MLHAT leverages on each incoming instance, updating sufficient statistics from the root to a leaf. For every attribute-value-class combination n_{ijk} , the algorithm tracks how often an attribute $X_i = x_{ij}$ appears with class y_k , enabling evaluation of potential splits. The base learner then chooses the best attribute using a heuristic function $G(\cdot)$, i.e., the information gain based on the *Bernoulli* distribution, where label priors are obtained from the counters maintained by the node [13]. It compares the top two attributes and applies the *Hoeffding Bound* to decide whether the difference ΔG is statistically significant, with confidence $1 - \delta_{tree}$:

$$\epsilon_m = \sqrt{\frac{\log_2(|L|)^2 \ln(1/\delta_{tree})}{2n}} \quad (6)$$

where L is the number of known label sets, δ is the desired confidence level, and n is the number of samples at the node m . A split occurs when $\Delta G > \epsilon_h$. In this way, the tree would expand incrementally as long as the entropy of the leaf nodes can be minimized with new splits. To manage complexity and prevent overfitting in incremental decision trees, two key hyperparameters are used: the *grace period* n_{min} , which defines how often splits are evaluated, and the *tie threshold* τ_{tree} , which serves as a pre-pruning mechanism. The tie threshold τ_{tree} limits growth when the difference $\Delta \bar{H}(X_i)$ between the best two attribute heuristics is small, i.e., $\Delta \bar{H}(X_i) \leq \tau_{tree}$. This helps avoid unstable decisions under noisy or ambiguous data. While a well-chosen τ mitigates unnecessary tree growth and improves scalability, its effectiveness is highly data-dependent. If too low, the tree may underfit; if too high, it risks overfitting and computational inefficiency. Together, τ_{tree} and n_{min} balance accuracy and complexity in data stream learning, allowing the production of increasingly larger trees that can divide the problem space into finer regions, better approximating the target concepts. However, such tree growth still remains highly uncontrollable, which can lead to two main issues: excessive memory consumption due to multiple redundant splits on all features, and a loss of plasticity due to the descendant nodes getting subsequently fixed to the space covered by their parent node. To address this, outdated subtrees are replaced with alternates that are grown when a drift detector detects drift through a changing error rather than on changing split attributes [5]. However, such split revision procedure can also reduce variance by simplifying the model, and introduce significant bias if important information is lost during subtree pruning [24]. Thus, only when the alternate subtree outperforms the original over a sufficient number of instances does it replace the latter. Following this intuition, MLHAT monitors the error in the Hamming loss, triggering replacements based on the *Hoeffding bound* ϵ_{alt} for alternating trees [5] and defined from monitored errors and instances seen by both main W and alternate W' sub-trees as:

$$\epsilon_{alt} = \sqrt{\frac{2e(1-e')(W+W') \ln(2/\delta_{alt-tree})}{W \cdot W'}} \quad (7)$$

where three scenarios can occur: (1) if $(e - e') > \delta_{alt-tree}$, the alternate tree replaces the main one, pruning the previous substructure resetting the drift detector in the alternate tree for future concept drifts, (2) if $(e' - e) > \delta_{alt-tree}$, the alternate tree remains without changes and the alternate tree is pruned, the concept drift was reversed, and (3) if $\delta_{alt-tree} \geq |e - e'|$, the two sub-trees continue to receive instances and expand to eventually differentiate their performance. This process aims to reduce bias and keep the model relevant by ensuring new subtrees adapt to the updated concept before deployment, with recursive replacements among alternates of alternates being possible. In other words, having alternative subtrees switch between training and test modes, and during the latter, also pruning the alternative subtrees whose accuracy does not increase over time. Moreover, leveraging on the fact that, in the midst of a concept drift, there will be multiple alternate sub-trees that are candidates to replace the branches whose performance is declining, their predictions is combined with the main tree, weighted based on the error monitored in each of the leaves involved, and conditioned on the alternate trees having received at least n_{min} instances. Thus, providing a good balance between the main structure that has the statistical guarantees of the *Hoeffding bound*, and a fast response to possible concept drifts. The full pseudocode for iHOMER is presented in Algorithm 1. *Note:* Parameters are scoped as follows: δ , and τ for trees use subscript `tree`; for label clustering, use `cluster`; and for alternate subtree comparisons, use `alt-tree`.

4 Experiments

iHOMER was compared against 19 other multi-label incremental algorithms proposed over the past 25 years, covering a broad spectrum of adaptability to both concept drift and class imbalance. Some of these algorithms address both challenges, while others focus on only one or neither. Within local models, binary relevance transformations of classical incremental decision tree models, like Hoeffding Tree (HT) [10], Hoeffding Adaptive Tree (HAT) [5], and Extremely Fast Decision Tree (EFDT) [24], were included, as well as more recent approaches such as Stochastic Gradient Tree (SGT) [18], Mondrian Tree (MT) [27], Aggregated Mondrian Forest (AMF) [27], and Adaptive Random Forest (ARF) [17]. Moreover, the comparison spans diverse families of algorithms, including Gaussian Naïve Bayes (NB) [6], k-Nearest Neighbors (kNN) [26], Adaptive Model Rules (AMR) [12], and ensemble methods like ADWIN Bagging of Logistic Regression (ABALR) [6] and ADWIN Boosting of Logistic Regression (ABOLR) [6]. Within global models, it also includes specialized multi-label methods like Multi-Label Hoeffding Tree (MLHT) [32], MLHT with Prune Set (MLHTPS) [32], Incremental Structured Output Prediction Tree (iSOOPT) [29], Multi-Label Broad Ensemble Learning System (MLBELS) [3], and Multi-Label Hoeffding Adaptive Tree (MLHAT) [13]. To enable a fair and reproducible comparison, the same hyperparameter configurations and evaluation protocols as those reported in the associated open-source repository referenced in the literature [13]. A detailed description of each algorithm's characteristics, along with the full parameter specifications used in the experiments, is available therein.

Evaluation metrics. Given the incremental nature of data streams, the standard prequential evaluation (also known as test-then-train [14]) was adopted. This approach evaluates the model on each instance before using it for training, allowing for continuous monitoring of predictive quality over time. Following prior work, we selected five widely-used multi-label metrics: **Subset Accuracy**, **Sample Accuracy**, **Hamming Loss**, **Micro F1**, and **Macro F1**. This choice

Algorithm 1 iHOMER

Given: n_{min} : minimum number of instances per node
 δ : confidence levels
 τ : tie-breaking thresholds
 $|L|$: number of known label sets, G : heuristic function
Initialize: cluster hierarchies $H \in \{H_{main}, H_{alt}\}$, each composed of models h

- 1: **for** each (x, y) in stream **do**
- 2: **for** each cluster hierarchy H of $\{H_{main}, H_{alt}\}$ **do**
- 3: Route x to clusters in H , and predict disjoint \hat{y}
- 4: // Online divisive agglomerative clustering
- 5: **for** each leaf l in H not yet tested **do**
- 6: Update $d(Jaccard)$
- 7: $\epsilon_l = \sqrt{\frac{R^2 \ln(1/\delta_{cluster})}{2n}}$
- 8: **if** $\frac{(d_1 - d_0)}{|d_1 + d_0 - 2d|} > \epsilon_l$ and $\epsilon_l < \tau_{cluster}$ **then**
- 9: Split leaf l
- 10: Initiate new clusters as pivot variables
- 11: Assign remaining based on pivot proximity
- 12: **end if**
- 13: **if** $2 \cdot \text{diam}(C_{parent}) - (\text{diam}(C_l) + \text{diam}(C_{sib})) \geq \epsilon_l$ **then**
- 14: Re-aggregate C_l and C_{sib} into C_{parent}
- 15: Reinitialize statistics at C_{parent}
- 16: Generate balanced structure
- 17: **end if**
- 18: **end for**
- 19: **for** each model h in cluster hierarchy H **do**
- 20: **for** each node n in the path to l **do**
- 21: Update node statistics with (x, y)
- 22: **if** alternate subtree $T_{alt}(n)$ exists **then**
- 23: Recursively update $T_{alt}(n)$ with (x, y)
- 24: **end if**
- 25: Update ADWIN with (\hat{y}, y)
- 26: **if** $T_{alt}(n)$ exists and $W(T_{alt}(n)) > n_{min}$ **then**
- 27: $\epsilon_{alt} = \sqrt{\frac{2e(1-e')(W+W') \ln(2/\delta_{alt-tree})}{W \cdot W'}}$
- 28: **if** $e - e' > \delta_{alt-tree}$ **then**
- 29: Replace main one with $T_{alt}(n)$
- 30: **else if** $e' - e > \delta_{alt-tree}$ **then**
- 31: Prune $T_{alt}(n)$
- 32: **end if**
- 33: **if** ADWIN warns and no $T_{alt}(n)$ exists **then**
- 34: $T_{alt}(n) \leftarrow$ new substitute subtree at n
- 35: **end if**
- 36: **if** n is a leaf l , and seen instances $> n_{min}$ **then**
- 37: Evaluate splits using heuristic G
- 38: $\epsilon_l = \sqrt{\frac{\log_2(|L|)^2 \ln(1/\delta_{tree})}{2n}}$
- 39: **if** $G_{best} - G_{sec} > \epsilon_l$ and $\epsilon_l < \tau_{tree}$ **then**
- 40: Split leaf l on best attribute
- 41: Create child nodes
- 42: **end if**
- 43: **end if**
- 44: **end for**
- 45: **end for**
- 46: Concatenate disjoint (\hat{y}, y) , and compute errors $\alpha_{main}, \alpha_{alt}$
- 47: **if** $\frac{\alpha_{main} - \alpha_{alt}}{\sqrt{\frac{\sigma_{main}^2}{N_{main}} + \frac{\sigma_{alt}^2}{N_{alt}}}} < t_{statistic}$ **then**
- 48: Replace the H_{main} with H_{alt}
- 49: **else if** drifting H_{main} **then**
- 50: (Re)initialize H_{alt}
- 51: **end if**
- 52: **end for**
- 53: **end for**
- 54: **end for**
- 55: **Output:** Final evaluation metrics

ensures a fair comparison with existing methods and captures both instance-level and label-level behavior. To analyze model behavior over time, the rolling Sample Accuracy was tracked using a sliding window approach. Subset Accuracy (also known as Exact Match) measures the proportion of instances where all predicted labels match exactly, whereas Sample Accuracy reflects partial correctness using the *Jaccard* index per instance. Let $Y_i = \{y_{i1}, \dots, y_{iL}\}$ and $Z_i = \{z_{i1}, \dots, z_{iL}\}$ be the true and predicted label sets for the i -th instance, respectively, with L labels and n instances, the metrics are:

$$\text{Subset Accuracy} = \frac{1}{n} \sum_{i=1}^n \mathbf{1}[Y_i = Z_i] \quad (8)$$

$$\text{Hamming Loss} = \frac{1}{nL} \sum_{i=1}^n \sum_{l=1}^L \mathbf{1}[y_{il} \neq z_{il}] \quad (9)$$

$$\text{Macro F1} = \frac{1}{L} \sum_{l=1}^L \frac{2 \cdot \sum_{i=1}^n \mathbf{1}[y_{il} = 1 \wedge z_{il} = 1]}{\sum_{i=1}^n \mathbf{1}[y_{il} = 1] + \sum_{i=1}^n \mathbf{1}[z_{il} = 1]} \quad (10)$$

$$\text{Micro F1} = \frac{2 \cdot \sum_{l=1}^L \sum_{i=1}^n \mathbf{1}[y_{il} = 1 \wedge z_{il} = 1]}{\sum_{l=1}^L \sum_{i=1}^n \mathbf{1}[y_{il} = 1] + \sum_{l=1}^L \sum_{i=1}^n \mathbf{1}[z_{il} = 1]} \quad (11)$$

Datasets. Table 1 provides a detailed overview of the real-world datasets used in this study, which are publicly available and widely used in multi-label classification tasks. These datasets range from small to large-scale, with up to 269,648 instances, 1836 features, and 374 labels, making them highly relevant for evaluating the scalability of iHOMER in high-dimensional streams. Additionally, the table includes key multi-label statistics such as *cardinality* (the average number of labels per instance), *density* (cardinality divided by the total number of labels), and the *mean imbalance ratio* (the average degree of label imbalance). The last column of the table indicates whether the instances in each dataset are presented in a temporal order: yes (✓), no (×), or unknown (-). While these datasets do not explicitly provide information about concept drift, the temporal ordering can offer insight into the potential for concept drift.

Table 1: Real-world datasets used. Temp. = Temporally ordered.

Dataset	Inst.	Feat.	Lbels	Card.	Dens.	MeanIR	Temp.
Flags	194	19	7	3.39	0.48	2.255	×
WaterQuality	1.060	16	14	5.07	0.36	1.767	-
Emotions	593	72	6	1.87	0.31	1.478	×
VirusGO	207	749	6	1.22	0.20	4.041	×
Birds	645	260	19	1.01	0.05	5.407	-
Yeast	2.417	103	14	4.24	0.30	7.197	×
Scene	2.407	294	6	1.07	0.18	1.254	×
CAL500	502	68	174	26.04	0.15	20.578	×
Human	3.106	440	14	1.19	0.08	15.289	-
Yelp	10.806	671	5	1.64	0.33	2.876	✓
Medical	978	1.449	45	1.25	0.03	89.501	✓
Eukaryote	7.766	440	22	1.15	0.05	45.012	-
Slashdot	3.782	1.079	22	1.18	0.05	19.462	✓
Hypercube	100.000	100	10	1.00	0.10	-	-
Hypersphere	100.000	100	10	2.31	0.23	-	-
Langlog	1.460	1.004	75	15.94	0.21	39.267	✓
Stackex	1.675	585	227	2.41	0.01	85.790	✓
Tmc	28.596	500	22	2.22	0.10	17.134	✓
Ohsumed	13.929	1.002	23	0.81	0.04	7.869	✓
D20ng	19.300	1.006	20	1.42	0.07	1.007	✓
Bibtex	7.395	1.836	159	2.40	0.02	12.498	×
Nuswidec	269.648	129	81	1.87	0.02	95.119	-
Imdb	120.919	1.001	28	1.00	0.04	25.124	×

Average results. Table 2 presents the average scores across four key evaluation metrics. Among all evaluated models, iHOMER achieves the best overall results on three of the four metrics—Subset Accuracy, Micro F1, and Hamming Loss—demonstrating its ability to adaptively model label dependencies through dynamic label clustering. This hybrid global-local architecture effectively captures co-occurrence patterns that static approaches miss. Compared to both global and local baselines, iHOMER shows consistent and substantial gains, particularly in Subset Accuracy (0.387) and Micro F1 (0.553). The closest competitor in Macro F1 is MLHAT, a global method, which outperforms iHOMER in that metric (0.445 vs. 0.282), likely due to a more balanced treatment of rare labels. In contrast, local models, which treat labels or label subsets independently, generally perform worse, especially on metrics sensitive to inter-label dependencies, such as Subset Accuracy and Micro F1. Nevertheless, local learners such as EFDT, HAT, kNN, and AMR emerged as the most competitive within their category. While these

Table 2: Avg. metric scores across datasets. Type = Global or Local.

Method	Subset Acc.	Micro F1	Macro F1	H. Loss	Type
iHOMER	0.387	0.553	0.282	0.093	G+L
MLHAT	0.351	0.550	0.445	0.105	G
MLBELS	0.281	0.502	0.326	0.123	G
MLHT	0.178	0.259	0.136	0.160	G
MLHTPS	0.175	0.298	0.208	0.132	G
iSOUP	0.269	0.436	0.303	0.114	G
KNN	0.275	0.430	0.303	0.111	L
NB	0.133	0.301	0.225	0.156	L
AMR	0.267	0.428	0.312	0.113	L
HT	0.204	0.385	0.257	0.116	L
HAT	0.240	0.423	0.287	0.110	L
EFDT	0.230	0.434	0.299	0.110	L
SGT	0.049	0.199	0.153	0.300	L
MT	0.171	0.288	0.188	0.123	L
ARF	0.260	0.417	0.288	0.098	L
AMF	0.243	0.403	0.281	0.101	L
ABALR	0.279	0.435	0.318	0.109	L
ABOLR	0.277	0.433	0.319	0.110	L

models demonstrate some capacity to model local patterns, they fall short when label relationships become complex or shift over time.

Critical difference diagrams. To validate the statistical significance of metric scores differences, the Friedman test was conducted, with the resulting statistics ranging between approximately 122 and 148 for all metrics. In all cases, the p-value was zero, indicating a clear rejection of the null hypothesis that all methods perform equally. A Nemenyi post-hoc test was then applied, with the resulting Critical Difference Diagrams (CDDs) for 95% confidence in Figures 5, 6, 7, and 8, further corroborating its consistent performance across all datasets. Notably, iHOMER ranks among the top three methods in all CDDs, underscoring its robustness across both label-based and instance-based performance criteria. In Subset Accuracy and Micro F1, iHOMER holds a statistically significant lead over most local learners and several global ones, affirming its effectiveness in capturing high-fidelity label combinations and optimizing overall predictive balance. While iHOMER remains highly competitive in Hamming Loss, it does not claim the top position. This outcome is partly attributable to the metric’s design: Hamming Loss penalizes every individual label misclassification equally, regardless of whether the rest of the label set is correctly predicted. As a hybrid model that prioritizes structured label dependencies and exact match accuracy, iHOMER optimizes for holistic correctness rather than minimizing marginal errors. Consequently, it achieves higher Subset Accuracy and F1 scores at the expense of slightly increased label-wise mismatches, which are more harshly reflected in Hamming Loss.

iHOMER vs. MLHAT. While MLHAT is recognized in the literature as a strong multi-label classifier, iHOMER consistently outperforms it across diverse scenarios involving varying label cardinality, density, and concept drift. These results highlight iHOMER’s ability to accurately model complex label dependencies and capture complete label combinations. As shown in Table 2, iHOMER achieves a substantial 10.26% improvement in Subset Accuracy (widely regarded as the most rigorous evaluation metric for multi-label tasks) and an 11.43% reduction in Hamming Loss compared to MLHAT. These gains indicate both higher predictive accuracy and fewer misclassifications. iHOMER also shows a modest improvement in Micro F1, while exhibiting slightly lower performance on Macro F1. This drop is expected, as iHOMER clusters disjoint label subsets, which may underrepresent rare labels when they are isolated into separate groups. Both models benefit from using multi-label metrics for drift detection, enabling quicker adaptation than methods that handle each label independently. Wilcoxon signed-rank tests confirm that the improvements in Subset Accuracy and Hamming Loss are statistically significant. Although iHOMER also achieves higher average Micro F1 scores, this particular difference does not reach statistical

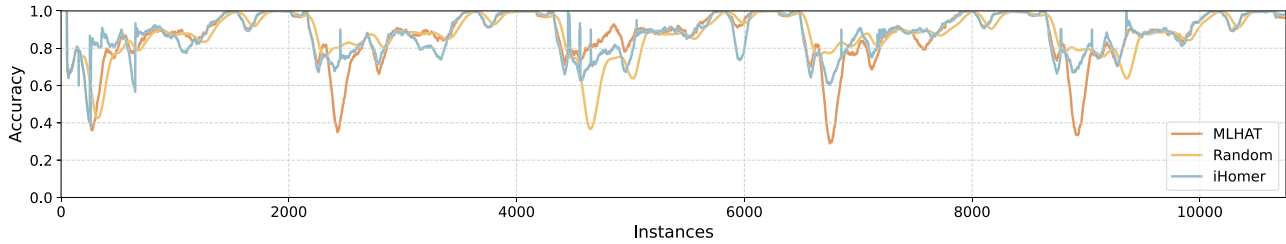


Figure 4: Rolling Sample Accuracy on the Yelp dataset.

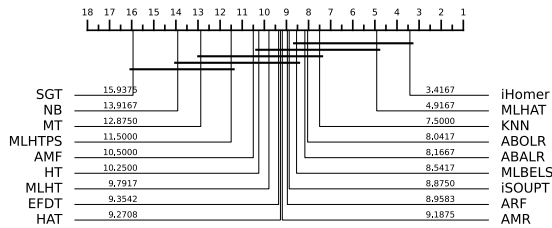


Figure 5: Nemenyi test - Subset Accuracy.

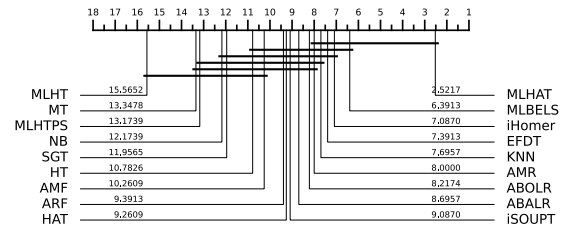


Figure 7: Nemenyi test - Macro F1.

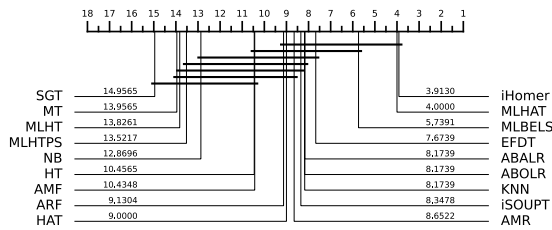


Figure 6: Nemenyi test - Micro F1.

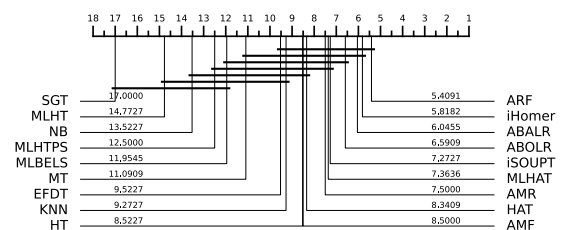


Figure 8: Nemenyi test - Hamming Loss.

significance. To illustrate iHOMER's superiority on a representative dataset, Figure 4 presents the rolling Sample Accuracy over time, highlighting the comparative performance of iHOMER, a single MLHAT, and MLHAT ensembles with randomly generated hybrid partitions. The rolling metric reveals that iHOMER produces smoother, more stable trends and avoids the sharp drops observed in other methods, underscoring the value of informed label space partitioning in streaming data scenarios.

Table 3: Wilcoxon signed-rank test: iHOMER vs. MLHAT.

Metric	Statistic	P-value	P-value < 0.05
Subset Acc.	72.0000	0.0249	✓
Micro F1	114.0000	0.4820	
Macro F1	21.0000	0.0001	✓
Hamming Loss	20.0000	0.0002	✓

5 Conclusions

In this paper, iHOMER (Incremental Hierarchy Of Multi-label Classifiers) was introduced for streaming multi-label learning. iHOMER incrementally partitions the label space into disjoint, correlated clusters without relying on predefined hierarchies. iHOMER leverages online divisive-agglomerative clustering based on *Jaccard* similarity and a global tree-based learner driven by a multivariate *Bernoulli* process to guide instance partitioning. To address non-stationarity, it integrates drift detection mechanisms at both global and local levels, enabling dynamic restructuring of label partitions and subtrees. Experiments across 23 benchmarks show that iHOMER outperforms state-of-the-art baselines by more than 20% and outperforms random search by 40%. These findings suggest that conventional multi-label streaming methods may not fully capture label correlations, and clus-

tering the label space offers a promising solution to addressing challenges such as high dimensionality, class imbalance, and the complexity of label correlations.

Future work. Despite its strengths, iHOMER still presents opportunities for improvement in computational efficiency. While it inherits MLHAT's ability to respond to concept drift via node resplitting and information gain approximation [13], as well as ODAC's dynamic dendrogram expansion based on cluster diameters [33], its growth efficiency remains suboptimal. Potential enhancements include dynamically tuning parameters such as *grace periods*, *tie thresholds*, and *split criteria* in response to incoming data. Further improvements could involve implementing stricter split evaluation rules (based on entropy, information gain, and minimum leaf instance counts), adaptive expansion strategies, and memory-aware mechanisms like leaf deactivation. These changes would allow more computational focus on frequently accessed nodes while reducing overhead elsewhere [22]. Finally, testing iHOMER on a broader range of datasets with annotated drift events could provide deeper insights into its adaptability and robustness.

Acknowledgements

Work funded by Portuguese Foundation for Science and Technology under project doi.org/10.54499/UIDP/00760/2020, Ph.D. scholarship doi.org/10.54499/PRT/BD/154713/2023, and Luso-American Development Foundation (FLAD Proj. 2025-0018). It also received EU funds, through Portuguese Republic's Recovery and Resilience Plan, within project PRODUTECH R3.

References

- [1] G. Aguiar, B. Krawczyk, and A. Cano. A survey on learning from imbalanced data streams: taxonomy, challenges, empirical study, and reproducible experimental framework. *Machine learning*, 113(7):4165–4243, 2024.
- [2] E. Almeida, C. Ferreira, and J. Gama. Adaptive model rules from data streams. In *ECML PKDD 2013*, pages 480–492. Springer, 2013.
- [3] S. Bakhshi and F. Can. Balancing efficiency vs. effectiveness and providing missing label robustness in multi-label stream classification. *Knowledge-Based Systems*, 289:111489, 2024.
- [4] M. Basgalupp, R. Cerri, L. Schietgat, I. Triguero, and C. Vens. Beyond global and local multi-target learning. *Information Sciences*, 579:508–524, 2021.
- [5] A. Bifet and R. Gavaldá. Adaptive learning from evolving data streams. In *IDA*, pages 249–260. Springer, 2009.
- [6] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldá. New ensemble methods for evolving data streams. In *15th ACM SIGKDD*, pages 139–148, 2009.
- [7] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771, 2004.
- [8] M. Breskvar, D. Kocev, and S. Džeroski. Ensembles for multi-target regression with random output selections. *Machine Learning*, 107:1673–1709, 2018.
- [9] A. Clare and R. D. King. Knowledge discovery in multi-label phenotype data. In *ECMLPKDD*, pages 42–53. Springer, 2001.
- [10] P. Domingos and G. Hulten. Mining high-speed data streams. In *Sixth ACM SIGKDD*, pages 71–80, 2000.
- [11] J. Duarte and J. Gama. Multi-target regression from high-speed data streams with adaptive model rules. In *2015 IEEE DSAA*, pages 1–10. IEEE, 2015.
- [12] J. Duarte, J. Gama, and A. Bifet. Adaptive model rules from high-speed data streams. *ACM TKDD*, 10(3):1–22, 2016.
- [13] A. Esteban, A. Cano, A. Zafra, and S. Ventura. Hoeffding adaptive trees for multi-label classification on data streams. *Knowledge-Based Systems*, 304:112561, 2024.
- [14] J. Gama, P. P. Rodrigues, and R. Sebastiao. Evaluating algorithms that learn from data streams. In *2009 ACM symposium on Applied Computing*, pages 1496–1500, 2009.
- [15] J. Gama, P. P. Rodrigues, E. Spinosa, and A. Carvalho. Knowledge discovery from data streams. In *Web Intelligence and Security*, pages 125–138. IOS Press, 2010.
- [16] E. C. Gatto, M. Ferrandin, and R. Cerri. Multi-label classification with label clusters. *Knowledge and Information Systems*, 67(2):1741–1785, 2025.
- [17] H. M. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck, B. Pfahringer, G. Holmes, and T. Abdesslem. Adaptive random forests for evolving data stream classification. *Machine Learning*, 106:1469–1495, 2017.
- [18] H. Gouk, B. Pfahringer, and E. Frank. Stochastic gradient trees. In *Asian conference on machine learning*, pages 1094–1109. PMLR, 2019.
- [19] J. Huang, G. Li, S. Wang, W. Zhang, and Q. Huang. Group sensitive classifier chains for multi-label classification. In *2015 ICME*, pages 1–6. IEEE, 2015.
- [20] J. D. C. Júnior, E. R. Faria, J. A. Silva, J. Gama, and R. Cerri. Novelty detection for multi-label stream classification under extreme verification latency. *Applied Soft Computing*, 141:110265, 2023.
- [21] D. Kocev, C. Vens, J. Struyf, and S. Džeroski. Ensembles of multi-objective decision trees. In *ECML 2007*, pages 624–631. Springer, 2007.
- [22] A. Lourenço, J. Rodrigo, J. Gama, and G. Marreiros. On-device edge learning for iot data streams: a survey. *arXiv preprint arXiv:2502.17788*, 2025.
- [23] G. Madjarov, D. Gjorgjević, and S. Džeroski. Two stage architecture for multi-label learning. *Pattern Recognition*, 45(3):1019–1034, 2012.
- [24] C. Manapragada, G. I. Webb, and M. Salehi. Extremely fast decision tree. In *24th ACM SIGKDD*, pages 1953–1962, 2018.
- [25] S. M. Mastelini, S. Barbon Jr, and A. C. P. d. L. F. de Carvalho. Online multi-target regression trees with stacked leaf models. *arXiv preprint arXiv:1903.12483*, 2019.
- [26] J. Montiel, M. Halford, S. M. Mastelini, G. Bolmier, R. Sourty, R. Vaysse, A. Zouitine, H. M. Gomes, J. Read, T. Abdesslem, et al. River: machine learning for streaming data in python. *Journal of Machine Learning Research*, 22(110):1–8, 2021.
- [27] J. Mourta, S. Gaïffas, and E. Scornet. Amf: Aggregated mondrian forests for online learning. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 83(3):505–533, 2021.
- [28] J. M. Moyano, E. L. Gibaja, K. J. Cios, and S. Ventura. Combining multi-label classifiers based on projections of the output space using evolutionary algorithms. *Knowledge-Based Systems*, 196:105770, 2020.
- [29] A. Osojnik, P. Panov, and S. Džeroski. Multi-label classification via multi-target regression on data streams. *Machine Learning*, 106:745–770, 2017.
- [30] Y. Papanikolaou, G. Tsoumakas, and I. Katakis. Hierarchical partitioning of the output space in multi-label data. *Data & Knowledge Engineering*, 116:42–60, 2018.
- [31] A. M. Ráez, L. A. U. López, and R. Steinberger. Adaptive selection of base classifiers in one-against-all learning for large multi-labeled collections. In *EsTAL 2004*, pages 1–12. Springer, 2004.
- [32] J. Read, A. Bifet, G. Holmes, and B. Pfahringer. Scalable and efficient multi-label classification for evolving data streams. *Machine learning*, 88:243–272, 2012.
- [33] P. Rodrigues, J. Gama, and J. P. Pedroso. Hierarchical clustering of time-series data streams. *Knowledge and Data Engineering, IEEE Transactions*, 20:615 – 627, 06 2008.
- [34] M. Roseberry and A. Cano. Multi-label knn classifier with self adjusting memory for drifting data streams. In *Second International Workshop on Learning with Imbalanced Domains: Theory and Applications*, pages 23–37. PMLR, 2018.
- [35] M. Roseberry, B. Krawczyk, and A. Cano. Multi-label punitive knn with self-adjusting memory for drifting data streams. *TKDD*, 13(6):1–31, 2019.
- [36] M. Roseberry, B. Krawczyk, Y. Djenouri, and A. Cano. Self-adjusting k nearest neighbors for continual learning from multi-label drifting data streams. *Neurocomputing*, 442:10–25, 2021.
- [37] M. Roseberry, S. Džeroski, A. Bifet, and A. Cano. Aging and rejuvenating strategies for fading windows in multi-label classification on data streams. In *38th ACM SIGAPP*, pages 390–397, 2023.
- [38] Z. Shi, Y. Wen, C. Feng, and H. Zhao. Drift detection for multi-label data streams based on label grouping and entropy. In *ICDM 2014*, pages 724–731. IEEE, 2014.
- [39] R. Sousa and J. Gama. Multi-label classification from high-speed data streams with adaptive model rules and random rules. *Progress in artificial intelligence*, 7(3):177–187, 2018.
- [40] E. Spyromitros-Xioufis, M. Spiliopoulou, G. Tsoumakas, and I. Vlahavas. Dealing with concept drift and class imbalance in multi-label stream classification. *Department of Computer Science, Aristotle University of Thessaloniki*, 2011.
- [41] B. Stevanoski, A. Kostovska, P. Panov, and S. Džeroski. Change detection and adaptation in multi-target regression on data streams. *Machine Learning*, 113(11):8585–8622, 2024.
- [42] P. Szymański, T. Kajdanowicz, and K. Kersting. How is a data-driven approach better than random choice in label space division for multi-label classification? *Entropy*, 18(8):282, 2016.
- [43] J. T. Tomás, N. Spolaór, E. A. Cherman, and M. C. Monard. A framework to generate synthetic multi-label datasets. *Electronic Notes in Theoretical Computer Science*, 302:155–176, 2014.
- [44] G. Tsoumakas, I. Katakis, and I. Vlahavas. Effective and efficient multilabel classification in domains with large number of labels. In *ECML/PKDD 2008*, volume 21, pages 53–59, 2008.
- [45] T. Wei, J.-X. Shi, and Y.-F. Li. Probabilistic label tree for streaming multi-label learning. In *27th ACM SIGKDD*, pages 1801–1811, 2021.
- [46] M.-L. Zhang and Z.-H. Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048, 2007.