Model-Agnostic Association Rule Learning with Tabular Foundation Models

Erkan Karabulut *e.karabulut@uva.nl*
University of Amsterdam
Vrije Universiteit Amsterdam

Daniel Daza *d.daza.cruz@vu.nl*
Vrije Universiteit Amsterdam

Paul Groth *p.t.groth@uva.nl*
University of Amsterdam

Martijn C. Schut *m.c.schut@amsterdamumc.nl*
Translational AI Laboratory, Department of Laboratory Medicine
Amsterdam University Medical Center, Vrije Universiteit Amsterdam

Victoria Degeler *v.o.degeler@uva.nl*
University of Amsterdam

Reviewed on OpenReview: *https://openreview.net/forum?id=Y7OlnNnA6T*

Abstract

Association Rule Mining (ARM) is a fundamental data mining technique for knowledge discovery in tabular data and is widely used in high-stakes decision-making. Existing approaches suffer from two limitations: reliance on frequent itemset mining leading to rule explosion in classical methods, and failure in low-data regimes in recent neural approaches. We reframe ARM as a conditional probability estimation problem and introduce a model-agnostic rule learning framework that extracts association rules from any conditional probabilistic model, decoupling rule extraction from frequent itemset mining. We introduce *TabProbe*, a probing-based algorithm that instantiates this framework, and demonstrate its generality across multiple model classes ranging from tree-based models to large pretrained foundation models. As a particularly powerful instantiation, we leverage tabular foundation models (TFMs), large models pretrained on diverse tabular data with strong in-context generalization, which can be applied *out-of-the-box* without task-specific training. Evaluation on 10 datasets of varying sizes shows that many framework instantiations achieve practically full data coverage with substantially fewer rules than state-of-the-art rule miners, and the best-performing instantiations achieve up to 15% higher association strength. Furthermore, TFM-learned rules have better predictive performance on unseen data and remain robust in low-data settings. Source code is available at https://github.com/DiTEC-project/tabprobe.

1 Introduction

Association Rule Mining (ARM) is an unsupervised knowledge discovery task that aims to find associations between features in a given dataset, such as tabular data, in the form of logical implications (Agrawal et al., 1994). It has been widely used across numerous domains for knowledge discovery (Kaushik et al., 2023; Luna et al., 2019) and high-stakes decision-making (Rudin, 2019) as part of interpretable Machine Learning (ML)

Table 1: **Sample clinical dataset and association rules.** Sample patient records from the Hepatitis dataset (Gong & Turney, 1983) with association rules extracted by our TabProbe from TabPFN (Hollmann et al., 2025), capturing interpretable patterns relating symptoms and treatments to patient survival.

Age	**Sex**	**Steroid**	**Antiviral**	$\cdots$	**Class**
30	female	no	yes	$\cdots$	LIVE
50	male	no	yes	$\cdots$	LIVE
78	male	yes	yes	$\cdots$	LIVE
39	male	yes	no	$\cdots$	DIE
Rule 1	Antiviral = yes $\wedge$ Malaise = yes $\rightarrow$ Class = LIVE				
Rule 2	Malaise = yes $\wedge$ Anorexia = yes $\rightarrow$ Class = LIVE				

models (Angelino et al., 2018). The ARM literature has been dominated by algorithmic methods that suffer from rule explosion and high computational overhead (Moens et al., 2013) without effective search space reduction (Srikant et al., 1997; Baralis et al., 2012; Yin et al., 2022; Fournier-Viger et al., 2012; Nguyen et al., 2018; Zaki & Hsiao, 2002). Such methods typically operate in two phases: mining frequent itemsets, and then generating association rules using the frequent itemsets.

Recent neurosymbolic methods, such as Aerial+ (Karabulut et al., 2025b), utilize autoencoders (Bank et al., 2023) to learn compact representations and extract small rule sets, overcoming rule explosion. However, such methods perform poorly on small datasets (Liu et al., 2017), leading to inaccurate rules that do not reflect true patterns (Karabulut et al., 2025a). This is particularly problematic as small tabular data is common in many domains, including biomedicine, such as rare disease datasets (Banerjee et al., 2023).

To address both rule explosion and low-data scenarios, we propose a novel framing of ARM as a conditional probability estimation problem and introduce a model-agnostic framework that formally decouples association rule extraction from frequent itemset mining, enabling any conditional probabilistic model over tabular data to learn association rules directly. We introduce *TabProbe*, a probing-based algorithm that instantiates this framework, and demonstrate its generality across multiple model classes, including tree-based models (XGBoost (Chen & Guestrin, 2016) and RandomForest (Breiman, 2001)) and large pretrained foundation models (Grinsztajn et al., 2026; Qu et al., 2025; Ma et al., 2024). As a particularly powerful instantiation, we leverage *Tabular Foundation Models (TFMs)* (Hollmann et al., 2025), large models pretrained on diverse tabular data with strong in-context generalization (Müller et al., 2022), which can be applied *out-of-the-box* without additional task-specific training, directly addressing both rule explosion and the low-data limitations of existing neural approaches. TFMs have demonstrated strong performance as classifiers for small tabular data by pretraining on large-scale synthetic (Qu et al., 2025) or real (Spinaci et al., 2025) tabular data and transferring learned representations to unseen datasets. Despite their growing success, TFMs have not yet been exploited for ARM. Samples from the clinical Hepatitis (Gong & Turney, 1983) dataset, together with association rules learned with TabPFN (Hollmann et al., 2025) using TabProbe, are shown in Table 1.

Our contributions are the following: (1) **A model-agnostic framework** for association rule learning, grounded in a novel framing of ARM as a conditional probability estimation problem, formally decoupling rule learning from frequent itemset mining and enabling any conditional probabilistic model to learn rules directly (Section 3); (2) **Two instantiations of the framework**: *TabProbe*, a novel probing-based algorithm for any single-target conditional probabilistic model, demonstrated with TFMs and tree-based models; and a multi-target instantiation for reconstruction-based models such as Autoencoders (Bank et al., 2023), demonstrating the generality of the framework across model classes (Section 3.1); (3) **A comprehensive evaluation** on 10 tabular datasets of small ($\sim$100 rows) and larger ($\geq$1000 rows) sizes in two settings: standard rule quality evaluation and downstream classification with cross-validation as part of interpretable rule-based classifiers (Liu et al., 1998; Angelino et al., 2018). With up to 7 times fewer rules, our framework instantiations achieve effectively full data coverage, up to 15% higher association strength (Yan et al., 2009) than state-of-the-art, better predictive performance on unseen data, and remain robust in low-data settings (Section 4).

2 Preliminaries and Related Work

Association Rule Mining. The ARM problem originally seeks rules of the form $X \rightarrow Y$ over categorical data that satisfy user-given minimum *support* and *confidence* thresholds (Agrawal et al., 1994). Support is the proportion of data instances containing $X \cup Y$. Confidence is the conditional frequency of Y among instances containing X.

Rule explosion. The ARM literature has been dominated by algorithmic methods, most notably the FP-Growth (Han et al., 2000) algorithm that has many variations, including parallel FP-Growth (Li et al., 2008) and FP-Growth on GPU (Jiang & Meng, 2017). Such methods work in two stages: finding frequent itemsets of higher support by co-occurrence counting and then forming rules of higher confidence based on given thresholds. However, the algorithmic methods suffer from rule explosion, that is, an excessive number of rules mined that are hard to interpret and post-process (Moens et al., 2013). This is due to the combinatorial growth in both the number of items, i.e., unique values, and their potential combinations into frequent itemsets and rules. The common approaches to overcome rule explosion in ARM include applying item constraints (Srikant et al., 1997; Baralis et al., 2012; Yin et al., 2022), mining only top-k rules (Fournier-Viger et al., 2012; Nguyen et al., 2018), closed itemset mining (Zaki & Hsiao, 2002), which refers to mining only frequent itemsets without frequent supersets of equal support, constraint programming (Belaid et al., 2019) and SAT-based enumeration (Izza et al., 2021). Beyond such search space reduction methods, recent neurosymbolic alternatives to addressing rule explosion leverage compact learned data representations.

Neurosymbolic methods in ARM. Despite the wide-scale success of deep learning, neural networks have only recently been applied to ARM. Aerial+ (Karabulut et al., 2025b) uses an undercomplete denoising autoencoder to learn lower-dimensional data representations and reconstruct the original inputs. Association rules are then derived based on reconstruction success, with the undercomplete structure favoring statistically significant rules over exhaustive enumeration. There have been other neural network-based approaches to ARM (Berteloot et al., 2024; Patel & Yadav, 2022). To the best of our knowledge, Aerial+ is the *only method* that explicitly addresses rule explosion. However, earlier work showed that in the small-data regime, Aerial+ suffers from insufficient training data, leading to inaccurate rules (Karabulut et al., 2025a).

Tabular foundation models. Prior-data fitted networks (PFNs) (Müller et al., 2022) pioneered training neural models on large collections of synthetic tabular tasks to enable in-context generalization and serve as a precursor to tabular foundation models. Tabular foundation models (TFMs) extend this paradigm by pretraining large-scale architectures on synthetic or real tabular data to learn transferable representations (Hollmann et al., 2025). Representative TFMs include TabPFN (Hollmann et al., 2025), trained on vast synthetic priors, TabICL (Qu et al., 2025), which broadens in-context learning across larger tables, and TabDPT (Ma et al., 2024), which leverages large-scale real tabular data. Despite the spreading success of TFMs across tabular tasks, use of TFMs to ARM has been limited to fine-tuning Aerial+ with embeddings from TabPFN (Karabulut et al., 2025a).

Our approach. While recent neurosymbolic methods address rule explosion using neural networks, they suffer from poor performance on small datasets, leading to rules that do not capture true patterns in the data. We reframe ARM as a conditional probability estimation problem and introduce a model-agnostic framework that formally decouples discovering association rules from frequent itemset mining. We demonstrate its generality across different model classes, ranging from tree-based models to large pretrained foundation models. Using this framing, we show that many tabular prediction models including TFMs can be used for categorical ARM *out-of-the-box* and that they address both rule explosion and low-data scenarios.

3 Model-Agnostic Rule Learning Framework

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a finite set of m items, and $D = \{t_1, t_2, \ldots, t_n\}$ a dataset of transactions, where each transaction $t_i \subseteq I$. For tabular data with k categorical features $F = \{f_1, \ldots, f_k\}$, each feature f_j

admits a finite set of c_j categories $\{f_j^1, \ldots, f_j^{c_j}\}$. The item universe is then

$$I = \bigcup_{j=1}^{k} \{f_j^1, \ldots, f_j^{c_j}\}, \tag{1}$$

and the number of items is $m = \sum_{j=1}^{k} c_j$. Each transaction can be represented as a one-hot encoded vector $\mathbf{x} \in \{0,1\}^m$ satisfying $\sum_{i=1}^{c_j} x[f_j^i] = 1$ for all $f_j \in F$.

Example. Table 1 shows a fragment of a clinical dataset where each patient record corresponds to a transaction containing items such as `Sex=male`, `Antiviral=yes`, or `Class=LIVE`. Rules such as `Antiviral=yes` $\wedge$ `Malaise=yes` $\rightarrow$ `Class=LIVE` capture interpretable associations between symptoms, treatments, and outcomes.

Definition 1 (Association Rule)**.** *An association rule is an implication $X \rightarrow Y$, where $X, Y \subseteq I$, is a first-order Horn clause with $|Y| = 1$, $|X| \geq 1$, and $X \cap Y = \varnothing$. The set X is referred as the antecedent and Y is the consequent.*

Classical rule *mining* methods enumerate frequent itemsets and apply thresholding on support and confidence, often leading to a combinatorial explosion of rules. In contrast, we consider association rule *learning* from trained probabilistic models that capture statistical regularities in the data.

Definition 2 (Conditional Probabilistic Model)**.** *Let $\mathcal{M}_\theta$ be a trained model that allows computing conditional queries of the form*

$$P_\theta(i \mid X), \quad i \in I,\ X \subseteq I, \tag{2}$$

interpreted as the model's estimated probability that item i holds given all items in X are observed to hold.

Example. Given the antecedent $X = \{$`Antiviral=yes`, `Malaise=yes`$\}$ from Table 1, the model $\mathcal{M}_\theta$ can be queried for $P_\theta($`Class=LIVE` $\mid X)$.

We *formalize* association rule learning from such models $\mathcal{M}_\theta$ via two requirements.

Requirement 1 (Antecedent Validation)**.** *A model $\mathcal{M}_\theta$ provides a scoring function $s_\theta : 2^I \rightarrow [0,1]$, where $s_\theta(X)$ quantifies how plausible the partial configuration X is under the learned data distribution. An antecedent X is considered valid if $s_\theta(X) \geq \tau_a$, for a user-specified threshold $\tau_a \in [0,1]$.*

Intuitively, this prevents the extraction of rules based on unlikely or spurious feature-value combinations.

Requirement 2 (Consequent Extraction)**.** *Given a valid antecedent X, the model allows computing conditional probabilities $P_\theta(i \mid X)$ for all $i \in I \setminus X$. An item i is accepted as a consequent of X if $P_\theta(i \mid X) \geq \tau_c$, for a user-specified threshold $\tau_c \in [0,1]$.*

Example. Let $X = \{$`Antiviral=yes`, `Malaise=yes`$\}$. Suppose the model assigns score $s_\theta(X) = 0.81$, and the antecedent threshold is set to $\tau_a = 0.6$, so that X is accepted as a valid antecedent. Querying the model for conditional probabilities yields:

$$P_\theta(\texttt{Class=LIVE} \mid X) = 0.92,\ P_\theta(\texttt{Class=DIE} \mid X) = 0.08.$$

With a consequent threshold of $\tau_c = 0.8$, the rule

$$\texttt{Antiviral=yes} \wedge \texttt{Malaise=yes} \rightarrow \texttt{Class=LIVE}$$

is therefore extracted by the framework.

Definition 3 (Association Validity Function)**.** *Given a model $\mathcal{M}_\theta$, the association validity function is*

$$\mathcal{V}_\theta(X \rightarrow Y) = \mathbf{1}[s_\theta(X) \geq \tau_a] \cdot \min_{i \in Y} P_\theta(i \mid X), \tag{3}$$

where $X, Y \subseteq I$ and $\mathbf{1}[\cdot]$ is an indicator function equal to one if its argument is true, and 0 otherwise.

Definition 4 (Association Rule Learning)**.** *Given a trained model $\mathcal{M}_\theta$ and thresholds $\tau_a, \tau_c \in [0,1]$, the association rule learning problem is to identify the set of rules*

$$\mathcal{R} = \{X \rightarrow Y \mid \mathcal{V}_\theta(X \rightarrow Y) \geq \tau_c\}. \tag{4}$$

Algorithm 1 Multi-Target Rule Learning

Require: Dataset D, model $\mathcal{M}_\theta$, thresholds τ_a, τ_c
1: $\mathcal{R} \leftarrow \emptyset$
2: **for all** antecedents of interest $X \subseteq I$ **do**
3: $\quad \mathbf{x} \leftarrow \text{ConstructProbe}(X)$
4: $\quad \hat{\mathbf{x}} \leftarrow \mathcal{M}_\theta(\mathbf{x})$
5: $\quad$ **if** $\hat{\mathbf{x}}[i'] \geq \tau_a, \forall i' \in X$ **then**
6: $\quad\quad Y \leftarrow \{i \in I \setminus X \mid \hat{\mathbf{x}}[i] \geq \tau_c\}$
7: $\quad\quad \mathcal{R} \leftarrow \mathcal{R} \cup \{X \rightarrow y \mid y \in Y\}$
8: **return** $\mathcal{R}$

3.1 Framework Instantiations

Requirements 1 and 2 can be satisfied through different model architectures. We demonstrate two paradigms, instantiated with multiple models, to show the generality of the framework across model classes.

3.1.1 Multi-Target Rule Learning Paradigm

We now show that multi-target predictive models, such as categorical autoencoders, naturally instantiate the proposed framework. In this setting, a model is trained to reconstruct, predict, or decode a data instance from a possibly corrupted, masked, or noisy input. A *multi-target instantiation* refers to any such model that simultaneously outputs predictions for all features and satisfies the framework requirements. Although not explicitly trained for rule learning, such models can be queried post hoc to extract rules by interpreting their outputs as conditional predictions under partial evidence.

Definition 5 (Multi-Target Instantiation)**.** *Let $\mathcal{M}_\theta : \mathbb{R}^m \rightarrow \mathbb{R}^m$ be a trained multi-target model that maps an input vector $\mathbf{x}$ to a reconstructed output $\hat{\mathbf{x}} = \mathcal{M}_\theta(\mathbf{x})$. We assume that the output is normalized per feature, such that for each feature $f_j \in F$, $\sum_{i \in f_j} \hat{\mathbf{x}}[i] = 1$, so that $\hat{\mathbf{x}}[i]$ is the predicted probability of item i.*

Given an antecedent set $X \subseteq I$, we construct an input probe vector $\mathbf{x}(X) \in \mathbb{R}^m$ where $\mathbf{x}(X)[i'] = 1$ for all $i' \in X$, $\mathbf{x}(X)[j] = 0$ for all other items j in features containing items from X, and $\mathbf{x}(X)[k] = 1/|f|$ for items k in remaining features f. We define the induced conditional probability as

$$P_\theta(i \mid X) := \mathcal{M}_\theta(\mathbf{x}(X))[i], \forall i \in I. \tag{5}$$

Example. Let $X = \{\texttt{Antiviral=yes}, \texttt{Malaise=yes}\}$. A sample input probe for $P_\theta(\texttt{Class=LIVE} \mid X)$ would be $p = x(X) = [1, 0, 1, 0, 0.5, 0.5]$, where $p[1], p[3] = 1$ *marks* items in X, $p[2], p[4] = 0$ marks other items in features containing X, and $p[5], p[6] = 0.5$ refers to remaining items. **Note** that uniform priors are one of several possible probing strategies for unmarked features, e.g., leaving them missing to indicate no prior knowledge, and the appropriate choice may depend on model architecture.

This instantiation satisfies the two requirements as follows.

- **Antecedent validation (Req. 1).** We define the antecedent score as $s_\theta(X) := \min_{i \in X} P_\theta(i|X)$, which quantifies how plausible the partial configuration X is under the learned data distribution.
- **Consequent extraction (Req. 2).** For a valid antecedent X, item $i \in I \setminus X$ is extracted as a consequent if $P_\theta(i \mid X) \geq \tau_c$.

Aerial+ (Karabulut et al., 2025b) instantiates this paradigm, allowing *simultaneous* antecedent validation and consequent extraction in a single forward pass, as the entire item space is reconstructed at once. **Algorithm 1** illustrates how multi-target association rule learning works in practice. Lines 2-7 iterate over antecedents of interest, e.g., all 3 combinations of $i \in I$. Line 3 constructs an input probe of size m as described earlier. Line 4 performs a forward pass to obtain reconstruction probabilities per feature. Lines 5-6 validate antecedents and extract consequents using thresholds τ_a and τ_c, and line 7 stores the rules.

Algorithm 2 TabProbe: Rule extraction from single-target models

Require: Dataset D, single-target model $\mathcal{M}_\theta$, max antecedents a, thresholds τ_a, τ_c

$\mathcal{R} \leftarrow \emptyset$, $F \leftarrow \text{GetFeatures}(D)$
$\mathbf{Q} \leftarrow \text{GenerateProbeVectors}(F, a)$, $\hat{\mathbf{P}} \leftarrow \mathbf{0}^{|\mathbf{Q}| \times m}$
for all feature $f_j \in F$ **do**
  $\mathbf{X}^{-j} \leftarrow \text{RemoveFeature}(D, f_j)$
  $\mathbf{y}^j \leftarrow \text{GetLabels}(D, f_j)$
  $\mathcal{M}_\theta.\text{fit}(\mathbf{X}^{-j}, \mathbf{y}^j)$
  $\mathbf{Q}^{-j} \leftarrow \text{RemoveFeature}(\mathbf{Q}, f_j)$
  $\hat{\mathbf{P}}[\cdot, f_j] \leftarrow \mathcal{M}_\theta.\text{predict_proba}(\mathbf{Q}^{-j})$
for all probe $\mathbf{q}_r \in \mathbf{Q}$ with antecedent set S_r **do**
  if $\min_{i \in S_r} \hat{\mathbf{P}}[r, i] \geq \tau_a$ **then**
    for all $i \in I \setminus S_r$ where $\hat{\mathbf{P}}[r, i] \geq \tau_c$ **do**
      $\mathcal{R} \leftarrow \mathcal{R} \cup \{S_r \rightarrow i\}$
return $\mathcal{R}$

3.1.2 Single-Target Rule Learning Paradigm

We now consider predictive models that are trained under a feature-wise single prediction objective, *single-target instantiation*, where each feature is predicted from the remaining ones. These models provide access to conditional predictions of the form $P_\theta(i \mid I \setminus f_j)$ for items i belonging to feature f_j. By aggregating such predictions across all features, we obtain a vector $\hat{\mathbf{x}}$ that serves as an approximation of the full conditional prediction vector required by the framework.

Definition 6 (Single-Target Instantiation)**.** *Let $\mathcal{M}_\theta$ be a predictive model that, given an input vector $\mathbf{x}$ and a target feature index j, outputs a probability distribution over the categories of feature f_j, $\mathcal{M}_\theta(\mathbf{x}, j) \in [0,1]^{|f_j|}$. To obtain a prediction vector over all items, we construct*

$$\hat{\mathbf{x}}[i] := \mathcal{M}_\theta(\mathbf{x}_{\setminus f_j}, j)[i], \quad \textit{for all } i \in f_j,\ j = 1, \ldots, k, \tag{6}$$

where $\mathbf{x}_{\setminus f_j}$ denotes the input with feature f_j masked or removed to prevent information leakage. Aggregating these predictions over all features yields a vector $\hat{\mathbf{x}} \in [0,1]^m$ that serves as an approximation of the full joint conditional prediction vector, consistent with the feature-wise prediction objective of single-target models.

Satisfaction of framework requirements. This instantiation satisfies the requirements as follows.

- **Antecedent validation (Req. 1).** As in the multi-target case, we define $s_\theta(X) := \min_{i \in X} \hat{x}[i]$, which quantifies how plausible the partial configuration X is under the learned data distribution.
- **Consequent extraction (Req. 2).** For a valid antecedent X, $i \in I \setminus X$ is a consequent if $\hat{\mathbf{x}}[i] \geq \tau_c$.

Example. Let $X = \{\texttt{Antiviral=yes}, \texttt{Malaise=yes}\}$ and target feature $f_j = \texttt{Class}$. Following Definition 6, we remove `Class` from D to prepare input matrix $\mathbf{X}^{-j}$, extract f_j labels as $\mathbf{y}^j$, and *fit* $\mathcal{M}_\theta$. The probe vector Q marks `Antiviral=yes` and `Malaise=yes` with probability 1 and assigns uniform priors to remaining features, similar to multi-target instantiation. Querying $\mathcal{M}_\theta$ with X yields $\hat{x}[\texttt{Class=LIVE}] = 0.92$. With $\tau_c = 0.8$, the rule $\texttt{Antiviral=yes} \wedge \texttt{Malaise=yes} \rightarrow \texttt{Class=LIVE}$ is extracted.

TFMs are a natural candidate for this paradigm as they excel at in-context prediction, accurately predicting features conditioned on others *without re-training*, and remain robust in low-data regimes. We instantiate the single-target formulation with TabPFNv3 (Grinsztajn et al., 2026), TabICLv2 (Qu et al., 2026), and TabDPT (Ma et al., 2024), as well as XGBoost (Chen & Guestrin, 2016), Random Forest (Breiman, 2001), canonical methods in tabular data, to demonstrate the generality of the framework across model classes.

Unlike multi-target instantiation, single-target predictive models do not allow explicit *marking* of antecedent features when predicting those same features, as this would introduce information leakage. This is an intentional design choice consistent with the feature-wise prediction objective of single-target models, such as TFMs or XGBoost, and its validity is supported by the empirical results presented in Section 4.

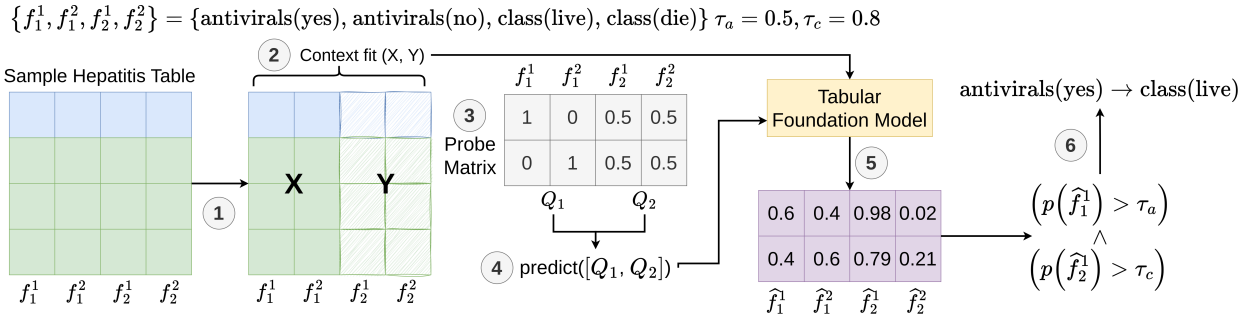


Figure 1: **Example** rule learning from TFMs with Algorithm 2: (1) split table into features (X) and labels (Y), (2) fit TFM on X and Y, (3) construct probing matrix with marked items (probability 1) and uniform priors elsewhere, (4) predict per feature, (5) aggregate predictions into reconstruction matrix, (6) apply thresholding to extract association rules.

Algorithm 2 illustrates how the single-target formulation can be realized with any single-target model $\mathcal{M}_\theta$. For a given max antecedent length a, line 2 uses *GenerateProbeVectors()* to build a matrix Q containing probe vectors for antecedent combinations of size up to a. Lines 3–8 iterate over each feature $f_j \in F$. *RemoveFeature()* removes f_j from D to prepare the input matrix $\mathbf{X}^{-j}$ (line 4). *GetLabels()* extracts f_j as target labels $\mathbf{y}^j$ (line 5). line 6 fits $\mathcal{M}_\theta$ on $(\mathbf{X}^{-j}, \mathbf{y}^j)$. *RemoveFeature()* removes f_j from Q to obtain $\mathbf{Q}^{-j}$ (line 7) and line 8 calls *predict_proba()* to obtain a probability distribution over values of f_j for probe vectors in Q at once and stores in $\hat{P}$. Lines 9–12 iterate over each probe $\mathbf{q}_r \in Q$ and its associated antecedent set S_r, perform antecedent validity (line 10) and consequent extraction (line 11) checks. Antecedent combinations and **probe vectors** are generated similarly to Algorithm 1. **Note** that *GenerateProbeVectors()* can also use iterative level-by-level probe generation as in Apriori, Aerial+, or other rule miners, restricting each antecedent length to combinations of items that already satisfied τ_a individually at shorter lengths. Figure 1 illustrates rule learning using Algorithm 2 from TFMs.

Time complexity. Ignoring the cost of fitting $\mathcal{M}_\theta$, which is dominated by the combinatorial terms as k grows (see **Appendix A** for a full analysis), Algorithm 2 has time complexity $O(k \cdot T_{\mathcal{M}_\theta}(|Q|) + k^{a+1} \cdot c_{\max}^a)$ in the worst-case, where $\mathcal{M}_\theta$ is queried once as a single batched forward pass per feature over the $|Q| = O(k^a \cdot c_{\max}^a)$ probing vectors from exhaustively enumerating and checking antecedent combinations of length up to a. k is the number of features, a the maximum antecedent size (constant, typically 2–5), and $c_{\max}$ the maximum categories per feature. This is the best achievable complexity for ARM with exhaustive antecedent enumeration, as ARM is inherently combinatorial. Since a is a constant, the complexity is polynomial.

4 Evaluation

This section evaluates our model-agnostic framework instantiations across 2 experimental settings: i) the standard rule quality evaluation procedure from the ARM literature, ii) downstream classification evaluation as part of interpretable machine learning models. Experiments are repeated for 10 random seeds. Setting 2 additionally applies 5-fold cross-validation, in line with standard practice in interpretable ML (Angelino et al., 2018; Liu et al., 1998). Setting 3 provides a scalability analysis. Appendix C provides extended results and additional analyses, and Appendix D provides a hyperparameter analysis.

Baselines. We initialize our model-agnostic framework with 6 different methods: TabPFNv3 (Grinsztajn et al., 2026), TabICLv2 (Qu et al., 2026), TabDPT (Ma et al., 2024), XGBoost (Chen & Guestrin, 2016) and RandomForest (Breiman, 2001) as single-target instantiations and Aerial+ (Karabulut et al., 2025b) as state-of-the-art neurosymbolic baseline, representative of the multi-target instantiation. Since our framework relies on the probabilistic interpretation of model outputs, we provide detailed model calibration diagnostics in Appendix B.4. FP-Growth (Han et al., 2000) is taken as one of the most popular algorithmic ARM baselines, implemented with MLxtend (Raschka, 2018). **Note** that the output of any other algorithmic

baseline is identical to that of FP-Growth (i.e., variants of Apriori (Agrawal et al., 1994), ECLAT (Zaki, 2002), or HMine (Pei et al., 2001)), as they are all deterministic, exhaustive enumerators. Furthermore, search space reduction methods from the ARM literature are structurally compatible with our framework, as our approach replaces co-occurrence counting rather than the search strategy.

Datasets. 5 small medical tabular datasets (∼100 rows) and 5 larger datasets (300 to 8000 rows) are used, summarized in Table 4 (Appendix B). All datasets have a class label to evaluate downstream classification as part of interpretable ML models (CBA (Liu et al., 1998) and CORELS (Angelino et al., 2018)). Numerical columns are discretized using equal-frequency discretization with 10 bins. For scalability analysis in Section 4.3, the large-scale canonical ARM benchmark T10I4D100K (Agrawal et al., 1994) is used.

Hyperparameters and fair comparison. To ensure a fair comparison, each method is run with a maximum antecedent length of 2. Framework instantiations are run with $\tau_c = 0.8$, and FP-Growth with a minimum confidence threshold of 0.8. In Setting 1, $\tau_a = 0.5$ for all instantiations, while FP-Growth is run with minimum support thresholds $\{0.5, 0.3, 0.2, 0.1\}$, and all results are reported. In Setting 2, $\tau_a = 0.3$ to obtain sufficient classification rules, which tend to have lower support; for the same reason, FP-Growth is run with minimum support thresholds $\{0.5, 0.3, 0.2, 0.1, 0.05, 0.01\}$. All TFMs are run with entire tables as context and an ensemble size of 8, without any tuning. XGBoost, RandomForest, and Aerial+ are tuned per dataset using Optuna (Akiba et al., 2019) (XGBoost, RandomForest) and grid search (Aerial+), respectively. FP-Growth has no learnable parameters and is run across all thresholds. The tuning reflects a standard practice: TFMs are designed for out-of-the-box use, while tree-based and autoencoder-based methods are tuned to their strongest reported configuration. Full parameter settings are listed in Table 7 (Appendix B).

4.1 Setting 1: Rule Quality Evaluation

Setup. In this experimental setting, we follow the standard ARM evaluation practice by computing 5 rule quality metrics on the full dataset, summarized in Table 5 (Appendix B). Zhang's metric measures association strength beyond statistical independence (i.e., non-spuriousness), and interestingness measures how unexpected a rule is relative to its antecedent and consequent supports.

Rule quality evaluation results. Table 2 shows that TFMs, XGBoost and Aerial+ achieve near-complete data coverage with up to 7 times fewer rules than FP-Growth, addressing rule explosion. Random forest achieves the highest confidence and association strength on small data, and highest confidence on larger data, with higher support than TFMs, XGBoost, and Aerial+, but at the cost of coverage (as low as 86.9%). TabPFNv3 stays closest to random forest's association strength on small datasets and surpasses it on larger datasets, while all TFMs retain near-complete coverage. Aerial+ shows lower support, association strength, and interestingness than TFMs and tree-based models on small datasets, consistent with its known degradation in low-data regimes. On larger datasets, Aerial+ recovers in confidence and interestingness, staying close to XGBoost in association strength but ∼15% below TFMs and random forest. Within TFMs, TabPFNv3 outperforms others in all metrics except being slightly behind in coverage. FP-Growth consistently produces lower association strength rules than TFMs and Random Forest, and requires lower support thresholds for full coverage, causing rule explosion. FP-G (0.5) has the highest interestingness and support on larger datasets but covers only 92.8% of the data with over 21,000 rules. The only metric where TFMs and XGBoost are consistently lower is confidence. However, as will be discussed below, TFM-learned rules of lower confidence outperform higher-confidence baseline rules on unseen data under 5-fold cross-validation in Setting 2, suggesting that classical methods overfit to dataset-specific co-occurrence frequencies. Figure 2 summarizes Table 2 via a normalized composite score $Q = \frac{1}{6}(\overline{\text{Conf}} + \overline{\text{Zhang}} + \overline{\text{Inter}}) + \frac{1}{2}\overline{\text{Cov}}$ weighted equally against Coverage since a rule set with poor coverage is of limited practical use regardless of its other quality, alongside rule counts (bars denote min-max normalization). Support is excluded as its optimal value is application-dependent. See **Appendix C.1** for per-dataset results.

Interpretation. Rule quality evaluation leads to the following core conclusions: i) predictive models can be repurposed to learn strong association rules as exemplified by 3 TFMs, XGBoost and random forest, ii) all 6 of our framework instantiations can address rule explosion by learning an order of magnitude smaller set of high-quality rules with full data coverage, iii) TabPFNv3 and random forest led to high association strength (Zhang) rules indicating strong non-redundancy, while the majority of the rules learned by algorithmic

Table 2: Average rule quality metrics (%), with ± denoting SD across seeds for non-deterministic methods. RandomForest has the highest confidence overall and the highest Zhang on small datasets, at the cost of coverage. All other instantiations achieve near-full coverage with up to 7 times fewer rules than FP-Growth (FP-G), which requires lower support thresholds (x) for full coverage. TabPFNv3 achieves the highest Zhang on larger and the second-highest on small datasets. Across-seed variation is higher on small datasets, and Cervical Cancer shows anomalously high variation across all non-deterministic methods (Table 11).

Datasets	Algorithm	#Rules	Support	Confidence	Zhang	Interestingness	Coverage
Small Tabular Datasets	TabICLv2	331	36.4 ± 3.4	82.9 ± 2.5	27.2 ± 8.5	40.8 ± 4.0	97.5 ± 3.1
	TabPFNv3	274	40.6 ± 1.8	86.5 ± 1.7	30.4 ± 6.3	44.8 ± 2.0	97.8 ± 1.1
	TabDPT	295	36.6 ± 3.3	82.9 ± 3.4	26.2 ± 8.3	40.9 ± 3.8	99.7 ± 0.6
	Aerial+	315	31.4 ± 2.8	87.5 ± 2.6	22.7 ± 4.3	37.1 ± 2.7	98.6 ± 1.9
	RandomForest	165	45.1 ± 2.2	92.8 ± 0.6	42.8 ± 4.6	49.7 ± 2.6	93.0 ± 1.7
	XGBoost	244	39.2	86.9	29.6	42.2	99.7
	FP-G (0.5)	152	57.8	92.4	23.8	63.2	86.3
	FP-G (0.3)	431	42.4	89.8	23.7	48.7	99.7
	FP-G (0.2)	680	34.5	90.9	24.4	41.1	100.0
	FP-G (0.1)	1463	22.9	91.7	26.0	29.2	100.0
Larger Tabular Datasets	TabICLv2	12481	48.2 ± 1.1	86.9 ± 1.2	34.5 ± 2.8	55.1 ± 1.7	99.2 ± 0.5
	TabPFNv3	11885	49.6 ± 0.9	88.3 ± 1.0	37.4 ± 2.3	57.2 ± 1.3	98.4 ± 0.7
	TabDPT	5177	45.8 ± 1.2	85.0 ± 1.4	36.4 ± 3.8	52.9 ± 1.8	98.3 ± 0.7
	Aerial+	7215	50.9 ± 2.3	93.6 ± 2.0	22.0 ± 2.6	57.5 ± 2.6	97.6 ± 2.1
	RandomForest	8126	57.8 ± 1.2	94.8 ± 0.6	36.3 ± 1.6	66.7 ± 1.6	86.9 ± 2.8
	XGBoost	9570	36.8	77.8	24.1	44.9	95.1
	FP-G (0.5)	21825	64.9	91.4	29.4	71.7	92.8
	FP-G (0.3)	26538	51.0	91.2	25.9	58.1	98.9
	FP-G (0.2)	29211	43.5	91.4	25.2	50.4	99.6
	FP-G (0.1)	35376	35.6	91.6	24.4	41.6	100.0

rule miners will be pruned out due to redundancy, iv) TFMs can address degraded performances in low-data scenarios, which affects the neurosymbolic baseline Aerial+, without sacrificing coverage as in random forest. **Why do TFMs show lower confidence but higher association strength?** Best-performing TFMs achieve 3–6% lower confidence than other baselines, while having comparable or higher association strength (Zhang) than all except random forest. High confidence in classical ARM methods reflects dataset-specific co-occurrence frequencies that do not necessarily generalize to unseen data, whereas association strength measures genuine feature associations beyond statistical independence. A rule capturing a spurious

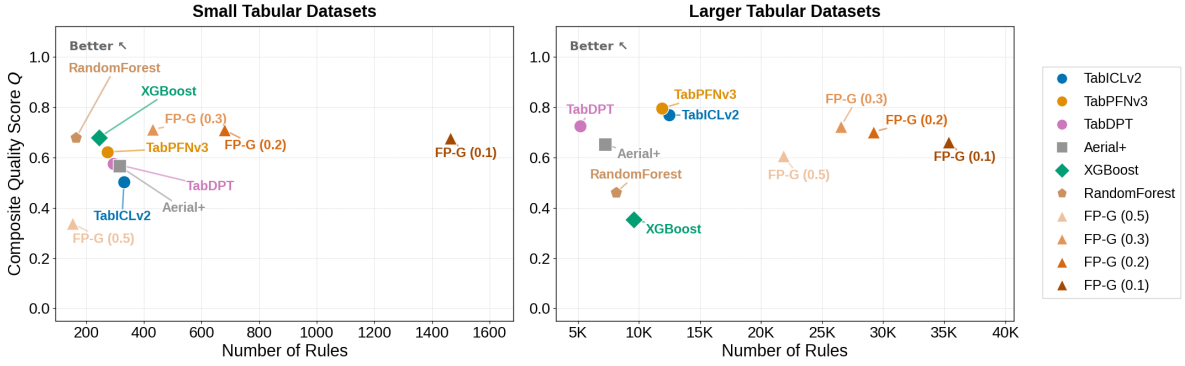


Figure 2: Composite rule quality score Q per method alongside rule counts. Tree-based models have the highest Q with a lower number of rules in small datasets, while TabPFNv3 achieves the highest Q among all baselines in larger datasets. As individual metrics capture different aspects of rule utility, Q should be interpreted as an exploratory summary rather than a definitive ranking.

correlation cannot systematically produce higher Zhang scores, as the metric controls for marginal statistics by construction. This is empirically confirmed in Setting 2, where TFM-learned rules with lower confidence consistently outperform higher-confidence baseline rules in 5-fold cross-validation, including random forests.

4.2 Setting 2: Downstream Task Evaluation

Setup. This experimental setting evaluates all methods as part of two well-known rule-based interpretable ML models: CBA (Liu et al., 1998; Filip & Kliegr, 2018) and CORELS (Angelino et al., 2018). The experimental pipeline is illustrated in Figure 5 (Appendix C.2). CBA and CORELS receive a large set of rules (CBA) or frequent itemsets (CORELS), and select a subset to build a rule-based classifier for high-stakes decision-making. We first learn rules or itemsets with each of the methods, pass those to CBA and CORELS to build a classifier, and measure accuracy, F1 score, precision, and recall. We use the same parameters as in Setting 1 except $\tau_a = 0.3$ for Aerial+, TFMs and tree-based models to ensure sufficient classification rules, with dataset-specific reductions detailed in Appendix B.3. For a fair comparison, FP-Growth is evaluated across minimum support thresholds $\{0.3, 0.2, 0.1, 0.01\}$, and all results are reported. The experiments are carried out for **10 random seeds** and with **5-fold cross-validation**, and the average results are presented. Algorithm 2 is adapted to mine frequent itemsets (Algorithm 3 of **Appendix E**) solely to accommodate CORELS, a canonical rule-based model, as frequent itemset mining is not a requirement of our method.

Results. Table 3 shows that on small datasets, TabDPT outperforms all baselines, while having higher variation across random seeds. TabICLv2 outperforms others on larger datasets with a small variation across seeds. TabPFNv3 is only within 1% behind TabICLv2 and TabDPT on small and larger datasets. Aerial+'s predictive performance is significantly lower in small tabular data, while it performs comparably to TFMs on larger datasets. The tree-based models are behind all other methods except Aerial+ and FP-G with high support thresholds on small tabular data. On larger datasets, tree-based model's performance increase, however, still behind all except TabPDT and FP-G (0.3). Variation across random seeds reduces on larger datasets for all methods. FP-Growth leads to higher predictive performance as the min support threshold decreases, at the expense of rule explosion, except FP-G (0.01) which does not lead to further performance gain. See Appendix C.2 for results per dataset and per classifier.

Interpretation. The best-performing TFM outperforms every other method, including all FP-Growth thresholds, on both small and larger datasets (TabDPT at 83.90% versus FP-G (0.1) at 83.54% on small data, and TabICLv2 at 88.31% versus FP-G (0.2) at 88.10% on larger data), supporting the validity of the model-agnostic framework. Against FP-Growth's most exhaustive configuration (0.01), TFMs match or exceed its accuracy using far fewer rules, e.g., TabICLv2 with 420 rules versus 2182 for FP-G (0.01) on larger data, indicating that additional rules beyond a certain point do not improve generalization. Tree-based instantiations confirm the framework extends beyond TFMs and autoencoders, though RandomForest and XGBoost remain behind all FP-Growth configurations on larger datasets, showing that not every instantiation is equally competitive across data regimes. TFM performance can be further improved by reducing τ_a to obtain more rules (see Appendix D for hyperparameter analysis).

4.3 Setting 3: Scalability Analysis

Setup. All methods are run for 5 random seeds with the same setup as Section 4.1, and average execution times are recorded. Experiments are conducted on the first 5,000 rows of the T10I4D100K dataset (Agrawal et al., 1994), a canonical benchmark for execution time evaluation in association rule mining, with an increasing number of items as the itemset universe is the decisive factor in scalability.

Results. Figure 3 shows that XGBoost, Random Forest, and Aerial+ are the fastest as the number of features increases, with XGBoost being the fastest of the three. TFMs consistently outperform all FP-Growth variants and scale to 200 columns within the 1,000-second limit, with TabDPT being the fastest TFM as the column count increases, while FP-Growth variants exceed this limit after 120–140 columns. TFMs achieve this despite being general-purpose foundation models orders of magnitude larger than, e.g.,

[1]Computed after excluding datasets where CBA extracted (near-)zero classification rules in $\geq$7/10 seeds. RandomForest failed to find rules on 5 datasets, and XGBoost on 2. Full per-dataset results are reported in Appendix C.2.

Table 3: Predictive performance (%) and average classification rule (CBA) or itemset (CORELS) count averaged across classifiers, with ± denoting standard deviation across random seeds. TabDPT outperforms all baselines on small, and TabICLv2 on larger datasets, using few rules. FP-Growth is deterministic, hence no variance (FP-G (x): minimum support x). Per-classifier and per-dataset results are in Appendix C.2.

Datasets	Method	#Rules	Accuracy	F1	Precision	Recall
Small Tabular Datasets	Aerial+	267	80.62 ± 1.87	77.33 ± 1.87	76.51 ± 2.04	80.62 ± 1.87
	TabPFNv3	145	82.66 ± 3.28	81.04 ± 3.25	81.21 ± 3.67	82.66 ± 3.28
	TabICLv2	165	83.32 ± 3.45	81.79 ± 3.53	82.09 ± 3.73	83.32 ± 3.45
	TabDPT	221	**83.90 ± 4.82**	**82.36 ± 4.94**	**82.56 ± 5.30**	**83.90 ± 4.82**
	XGBoost[1]	156	79.48 ± 2.39	77.29 ± 2.37	77.59 ± 2.76	79.48 ± 2.39
	RandomForest[1]	125	81.12 ± 2.21	79.01 ± 2.34	78.71 ± 2.74	81.12 ± 2.21
	FP-G (0.3)	57	79.72	76.45	75.84	79.72
	FP-G (0.2)	93	79.95	77.08	76.86	79.95
	FP-G (0.1)	221	83.54	81.37	82.49	83.54
	FP-G (0.01)	1916	83.28	80.58	81.16	83.28
Larger Tabular Datasets	Aerial+	754	87.56 ± 0.78	86.13 ± 0.87	86.34 ± 1.01	87.56 ± 0.78
	TabPFNv3	432	87.86 ± 1.48	87.16 ± 1.41	87.46 ± 1.53	87.86 ± 1.48
	TabICLv2	420	**88.31 ± 0.88**	**87.64 ± 0.91**	**88.02 ± 1.00**	**88.31 ± 0.88**
	TabDPT	410	84.94 ± 2.73	84.32 ± 2.75	84.82 ± 2.79	84.94 ± 2.73
	XGBoost[1]	695	84.72 ± 2.12	83.38 ± 2.13	83.46 ± 2.09	84.72 ± 2.12
	RandomForest[1]	423	84.10 ± 3.57	83.50 ± 3.59	84.66 ± 3.77	84.10 ± 3.57
	FP-G (0.3)	292	84.90	84.36	84.53	84.90
	FP-G (0.2)	385	88.10	87.30	87.30	88.10
	FP-G (0.1)	617	87.87	86.50	86.33	87.87
	FP-G (0.01)	2182	88.04	86.60	86.41	88.04

the specialized model of Aerial+. This demonstrates that the scalability concern associated with exhaustive enumeration is more severe in practice for standard rule miners than TFM-based rule learning.

Interpretation. XGBoost, Random Forest, and Aerial+ achieve the fastest execution times, ahead of TFMs, since tree-based models and Aerial+'s compact autoencoder are far smaller and cheaper to fit than a general-purpose foundation model. This, combined with the rule quality and generalization performance demonstrated in Settings 1 and 2, means the choice between TFMs and others involves a tradeoff between speed and rule quality. TFMs, Aerial+ and tree-based models alike scale beyond FP-Growth variants, demonstrating that the scalability concern associated with exhaustive enumeration is more severe in practice. Considering the speed of advancement in TFMs, going from tables of up to 10,000 rows (Hollmann et al.,

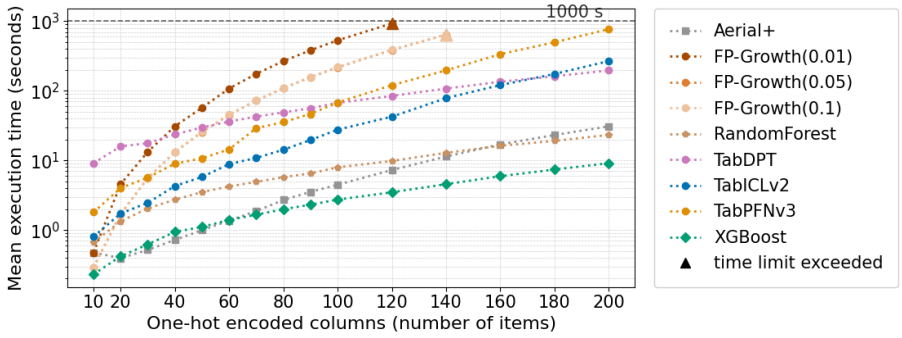


Figure 3: Mean execution time on the canonical T10I4D100K ARM benchmark as the number of columns increases. XGBoost, RandomForest, and Aerial+ are the fastest, and TabDPT is the fastest TFM. All TFMs scale to 200 columns within the 1,000-second limit, faster than FP-Growth, which exceeds the limit after 120–140 columns, despite being orders of magnitude larger (FP-G (x): min support threshold).

2022; 2025) to 500,000 rows and 2,000 columns (Ma et al., 2024) within 2 years, execution time is expected to improve further, independently of our framework.

5 Discussion

Model-agnostic association rule learning. The framework introduced in Section 3 reframes ARM as a learning problem rather than a search over co-occurrence counts, shifting the question from *how often do these items appear together?* to *what does the learned model believe about their conditional dependency?* This new framing applies to any conditional probabilistic model on tabular data satisfying two requirements: antecedent validation, which checks whether a candidate antecedent configuration is plausible under the learned data distribution, and consequent extraction, which identifies items whose conditional probability exceeds a threshold given that antecedent. Together, these two requirements decouple rule extraction from frequent itemset mining, enabling any model that can estimate conditional probabilities to serve as the backbone of an ARM system. As demonstrated with TFMs, tree-based models, and reconstruction-based models in Section 3.1, a diverse range of model classes can instantiate the framework, offering practitioners flexibility in choosing a model suited to their data and computational constraints. This framework also opens a new avenue for neural network interpretability, as association rules provide inherently interpretable explanations of learned patterns in an otherwise opaque model.

Tabular foundation models can learn association rules. TFMs with a classification objective can be adapted to learn association rules using the single-target instantiation of our framework out-of-the-box without re-training or architecture optimization, in contrast to neurosymbolic methods such as Aerial+, which require per-dataset training and architecture fine-tuning. As shown in Setting 1 (Section 4.1), TFMs produce concise rule sets with effectively full data coverage, addressing rule explosion in ARM on tabular data. Furthermore, TFM-learned rules show stronger generalizability than state-of-the-art methods, achieving higher association strength and better downstream performance on unseen data under cross-validation, as confirmed in Setting 2 (Section 4.2). Notably, lower-confidence TFM rules consistently outperform higher-confidence baseline rules on held-out data, suggesting that classical methods overfit to dataset-specific co-occurrence frequencies while TFMs capture genuine feature associations that generalize to unseen data.

Scalability. As demonstrated in Setting 3 (Section 4.3), all framework instantiations scale beyond FP-Growth in execution time, reaching 200 items within the 1,000-second limit while FP-Growth variants exceed it beyond 120–140 items. As a widely adopted and representative algorithmic baseline, FP-Growth reflects the scalability of co-occurrence counting methods in general; the superior scaling of our framework instantiations therefore suggests a fundamental advantage of the learning-based framing over counting-based ARM as the itemset universe grows. Tree-based models and Aerial+ achieve the fastest execution times, while TFMs remain within the time limit despite being general-purpose foundation models orders of magnitude larger than these specialized methods. Feature-wise predictions in TFMs are parallelizable across features, and some TFMs (e.g., TabPFNv3) support context caching to further speed up inference. The remaining scalability constraint is the context size limit of current TFM architectures, although this has improved dramatically, from tables of up to 10,000 rows (Hollmann et al., 2022) to 500,000 rows (Qu et al., 2025) within two years, and is expected to continue improving independently of our framework.

Practical implications. Many concepts from the ARM literature that reduce search space are structurally compatible with our framework and Algorithms 1 and 2, though pruning guarantees relying on the anti-monotonicity of exact support may not carry over to model-estimated scores. These include ARM with item constraints (Yin et al., 2022), where probing vectors are constructed for features of interest only, and top-k rule mining (Fournier-Viger et al., 2012), where rules of top-k highest probability are learned rather than all rules above a threshold. Probing vectors can also be constructed iteratively, limited to features exceeding τ_a in the previous iteration as a heuristic approximation, reducing the probing matrix size and computational cost. Probing strategies tailored to specific model architectures are also a promising direction for future work. As shown in the hyperparameter analysis in Appendix D, τ_a and τ_c allow practitioners to balance rule set size and rule quality according to their needs, with lower τ_a favoring recall and higher τ_c favoring precision in downstream classification (see also Appendices C.6 and C.2 for sensitivity to antecedent length and binning resolution). For practitioners without GPU access or large pretrained models, the framework

can be instantiated with any lightweight conditional probabilistic model, as demonstrated with tree-based models (Section 4.1). Appendix C.8 provides further guidance on model selection.

6 Conclusions and Future Work

This paper introduced a model-agnostic framework that reframes association rule mining as a conditional probability estimation problem, formally decoupling rule learning from frequent itemset mining. The framework applies to any conditional probabilistic model over tabular data that satisfies two requirements: a scoring mechanism for antecedent validation and conditional probability estimation for consequent extraction. We instantiate the framework with tabular foundation models, tree-based models, and reconstruction-based models, demonstrating its generality across model classes.

In two comprehensive experimental settings on 10 datasets, we demonstrated that all framework instantiations, except RandomForest on larger datasets, achieve practically full data coverage with substantially fewer rules than classical methods, with the best-performing TFM having up to 15% higher association strength, while tree-based instantiations further confirm the framework's generality. These rules also generalize to unseen data, achieving equal or higher classification performance within interpretable machine learning pipelines despite using far fewer rules.

This work opens several directions for future research. Evaluating other conditional probabilistic models within the framework is a natural next step. Extending the approach to numerical data without prediscretization, adapting it to non-tabular structures such as graphs, and applying it to various association rule mining variants, such as sequential rule mining and high-utility itemset mining, are further promising directions. More broadly, our framework shows how pretrained foundation models can be leveraged for interpretable knowledge discovery without task-specific training.

Acknowledgment. This work has received support from the Dutch Research Council (NWO), in the scope of the Digital Twin for Evolutionary Changes in water networks (DiTEC) project, file number 19454.

References

Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB*, volume 1215, pp. 487–499, 1994.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631, 2019.

Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. Learning certifiably optimal rule lists for categorical data. *Journal of Machine Learning Research*, 18(234):1–78, 2018.

Jineta Banerjee, Jaclyn N Taroni, Robert J Allaway, Deepashree Venkatesh Prasad, Justin Guinney, and Casey Greene. Machine learning in rare disease. *Nature Methods*, 20(6):803–814, 2023.

Dor Bank, Noam Koenigstein, and Raja Giryes. Autoencoders. *Machine learning for data science handbook: data mining and knowledge discovery handbook*, pp. 353–374, 2023.

Elena Baralis, Luca Cagliero, Tania Cerquitelli, and Paolo Garza. Generalized association rule mining with constraints. *Information Sciences*, 194:68–84, 2012.

Mohamed-Bachir Belaid, Christian Bessiere, and Nadjib Lazaar. Constraint programming for association rules. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pp. 127–135. SIAM, 2019.

Théophile Berteloot, Richard Khoury, and Audrey Durand. Association rules mining with auto-encoders. In *International Conference on Intelligent Data Engineering and Automated Learning*, pp. 51–62. Springer, 2024.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983.

Jiří Filip and Tomáš Kliegr. Classification based on associations (cba)-a performance analysis. Technical report, EasyChair, 2018.

Philippe Fournier-Viger, Cheng-Wei Wu, and Vincent S Tseng. Mining top-k association rules. In *Advances in Artificial Intelligence: 25th Canadian Conference on Artificial Intelligence, Canadian AI 2012, Toronto, ON, Canada, May 28-30, 2012. Proceedings 25*, pp. 61–73. Springer, 2012.

Ashish Ghosh and Bhabesh Nath. Multi-objective rule mining using genetic algorithms. *Information Sciences*, 163(1-3):123–133, 2004.

W Brier Glenn et al. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78 (1):1–3, 1950.

Gail Gong and Peter Turney. Hepatitis dataset, 1983. URL `https://archive.ics.uci.edu/dataset/46/hepatitis`.

Gösta Grahne and Jianfei Zhu. Efficiently using prefix-trees in mining frequent itemsets. In *FIMI*, volume 90, pp. 65, 2003a.

Gösta Grahne and Jianfei Zhu. High performance mining of maximal frequent itemsets. In *6th International workshop on high performance data mining*, volume 16, pp. 34, 2003b.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin, Magnus Bühler, Anurag Garg, et al. Tabpfn-3: Technical report. *arXiv preprint arXiv:2605.13986*, 2026.

Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. *ACM sigmod record*, 29(2):1–12, 2000.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Yacine Izza, Saïd Jabbour, Badran Raddaoui, and Abdelhamid Boudane. On the enumeration of association rules: A decomposition-based approach. In *Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI 2020)*, pp. 1265–1271, 2021.

Hao Jiang and He Meng. A parallel fp-growth algorithm based on gpu. In *2017 IEEE 14th International Conference on e-Business Engineering (ICEBE)*, pp. 97–102. IEEE, 2017.

Erkan Karabulut, Daniel Daza, Paul Groth, and Victoria Degeler. Discovering association rules in high-dimensional small tabular data. In *Proceedings of the 1st International Workshop on Advanced Neuro-Symbolic Applications (ANSyA) at the European Conference on Artificial Intelligence (ECAI 2025)*, 2025a. Accepted/In press.

Erkan Karabulut, Paul Groth, and Victoria Degeler. Neurosymbolic association rule mining from tabular data. In *Proceedings of The 19th International Conference on Neurosymbolic Learning and Reasoning (NeSy 2025)*, volume 284 of *Proceedings of Machine Learning Research*, pp. 565–588. PMLR, 08–10 Sep 2025b. URL `https://proceedings.mlr.press/v284/karabulut25a.html`.

Erkan Karabulut, Paul Groth, and Victoria Degeler. Pyaerial: Scalable association rule mining from tabular data. *SoftwareX*, 31:102341, 2025c. ISSN 2352-7110. doi: https://doi.org/10.1016/j.softx.2025.102341.

Minakshi Kaushik, Rahul Sharma, Iztok Fister Jr, and Dirk Draheim. Numerical association rule mining: a systematic literature review. *arXiv preprint arXiv:2307.00662*, 2023.

Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. The uci machine learning repository. 2023. URL `https://archive.ics.uci.edu`.

Haoyuan Li, Yi Wang, Dong Zhang, Ming Zhang, and Edward Y Chang. Pfp: parallel fp-growth for query recommendation. In *Proceedings of the 2008 ACM conference on Recommender systems*, pp. 107–114, 2008.

Bing Liu, Wynne Hsu, and Yiming Ma. Integrating classification and association rule mining. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, KDD*, pp. 80–86, 1998.

Bo Liu, Ying Wei, Yu Zhang, and Qiang Yang. Deep neural networks for high dimension, low sample size data. In *IJCAI*, volume 2017, pp. 2287–2293, 2017.

José María Luna, Philippe Fournier-Viger, and Sebastián Ventura. Frequent itemset mining: A 25 years review. *WIREs Data Mining and Knowledge Discovery*, 9(6):e1329, 2019. doi: https://doi.org/10.1002/widm.1329. URL `https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1329`.

Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Hamidreza Kamkari, Alex Labach, Jesse C Cresswell, Keyvan Golestan, Guangwei Yu, Maksims Volkovs, and Anthony L Caterini. Tabdpt: Scaling tabular foundation models. *arXiv preprint arXiv:2410.18164*, 2024.

Sandy Moens, Emin Aksehirli, and Bart Goethals. Frequent itemset mining for big data. In *2013 IEEE international conference on big data*, pp. 111–118. IEEE, 2013.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=KSugKcbNf9`.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

Linh TT Nguyen, Bay Vo, Loan TT Nguyen, Philippe Fournier-Viger, and Ali Selamat. Etarm: an efficient top-k association rule mining algorithm. *Applied Intelligence*, 48:1148–1160, 2018.

Harvendra Kumar Patel and KP Yadav. An innovative approach for association rule mining in grocery dataset based on non-negative matrix factorization and autoencoder. *Journal of Algebraic Statistics*, 13 (3):2898–2905, 2022.

Jian Pei, Jiawei Han, Hongjun Lu, Shojiro Nishio, Shiwei Tang, and Dongqing Yang. H-mine: Hyper-structure mining of frequent patterns in large databases. In *proceedings 2001 IEEE international conference on data mining*, pp. 441–448. IEEE, 2001.

John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. Tabicl: A tabular foundation model for in-context learning on large data. *arXiv preprint arXiv:2502.05564*, 2025.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. Tabiclv2: A better, faster, scalable, and open tabular foundation model. *arXiv preprint arXiv:2602.11139*, 2026.

Sebastian Raschka. Mlxtend: Providing machine learning and data science utilities and extensions to python's scientific computing stack. *The Journal of Open Source Software*, 3(24), April 2018. doi: 10.21105/joss.00638.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

Marco Spinaci, Marek Polewczyk, Maximilian Schambach, and Sam Thelin. Contexttab: A semantics-aware tabular in-context learner. *arXiv preprint arXiv:2506.10707*, 2025.

Ramakrishnan Srikant, Quoc Vu, and Rakesh Agrawal. Mining association rules with item constraints. In *Kdd*, volume 97, pp. 67–73, 1997.

Xiaowei Yan, Chengqi Zhang, and Shichao Zhang. Confidence metrics for association rule mining. *Applied Artificial Intelligence*, 23(8):713–737, 2009.

Zhaowen Yin, Wensheng Gan, Gengsen Huang, Yongdong Wu, and Philippe Fournier-Viger. Constraint-based sequential rule mining. In *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–10. IEEE, 2022.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 694–699, 2002.

Mohammed J Zaki and Ching-Jui Hsiao. Charm: An efficient algorithm for closed itemset mining. In *Proceedings of the 2002 SIAM international conference on data mining*, pp. 457–473. SIAM, 2002.

Mohammed Javeed Zaki. Scalable algorithms for association mining. *IEEE transactions on knowledge and data engineering*, 12(3):372–390, 2002.

A Runtime Complexity Analysis

This section provides a runtime complexity analysis of Algorithm 2 in big O notation.

Initials. Building on the ARM problem definition given in Section 2 and Definition 1; let k denote the number of features in a tabular data, $c_{\max} = \max_j c_j$ the maximum categories per feature, $m = \sum_{j=1}^{k} c_j$ the total number of items, n the number of transactions in D, a the maximum antecedent size, and $T_{\mathcal{M}_\theta}(n)$ the time complexity of *fitting* and *predicting* with tabular foundation model $\mathcal{M}_\theta$ on n rows.

Component-wise analysis. We analyze each component of the algorithm separately and then aggregate the results.

Outer Loops. Line 2 generates $\binom{k}{i}$ feature combinations for each antecedent size i from 1 to a in a single call to *GenerateProbeVectors()*, and generates a total of

$$\sum_{i=1}^{a} \binom{k}{i} = O(k^a) \quad \text{combinations for constant } a \tag{7}$$

Probe Vector Generation. For each feature combination S with $|S| = i$ generated in line 2, we generate probing vectors for all feature category combinations. The number of probing vectors is $\prod_{f \in S} c_f \leq c_{\max}^i$. Each probe has a dimension m. Summing over all combinations, the total number of probing vectors is as follows:

$$|Q| = \sum_{i=1}^{a} \binom{k}{i} \cdot c_{\max}^i = O(k^a \cdot c_{\max}^a) \quad \text{for constant } a \tag{8}$$

Therefore, generating the probing matrix Q takes $O(|Q| \cdot m)$.

Feature-wise Prediction. Lines 3 to 8 iterates over the k features and performs the followings:

- Prepare X_{ctx} context by removing feature f_j from D in line 4, and store it as y_{ctx} in line 5. This takes $O(1)$.
- Fit the foundation model in line 6 which takes $T_{\mathcal{M}_\theta}^{\text{fit}}(n)$.
- Remove f_j from Q in line 7 and predict on all remaining Q^{-j} probing vectors in line 8, which takes $T_{\mathcal{M}_\theta}^{\text{pred}}(|Q|)$.

The total complexity of feature-wise predictions for k features becomes $O(k \cdot (T_{\mathcal{M}_\theta}(n) + T_{\mathcal{M}_\theta}(|Q|)))$.

Rule Extraction. Lines 9 to 12 extract rules from the final prediction matrix $\hat{P}$ by checking each k features for $|Q|$ probes, performing thresholding, in $O(k \cdot |Q|)$ time.

Total Complexity. Aggregating complexities across all components:

$$T_{\text{total}} = k \cdot (T_{\mathcal{M}_\theta}(n) + T_{\mathcal{M}_\theta}(|Q|)) + k \cdot |Q| \tag{9}$$

Substituting $|Q| = O(k^a \cdot c_{\max}^a)$ for constant a gives

$$T_{\text{total}} = O\left(k \cdot T_{\mathcal{M}_\theta}(n) + k \cdot T_{\mathcal{M}_\theta}(|Q|) + k^{a+1} \cdot c_{\max}^a\right) \tag{10}$$

Thus, Algorithm 2 has time complexity $O(k \cdot T_{\mathcal{M}_\theta}(n) + k \cdot T_{\mathcal{M}_\theta}(|Q|) + k^{a+1} \cdot c_{\max}^a)$ where k is the number of features, a is the maximum antecedent size, n is the dataset size, and $c_{\max}$ is the maximum classes per feature. Since $k^{a+1} \cdot c_{\max}^a$ grows in k for $a \geq 1$ faster than $k \cdot T_{\mathcal{M}_\theta}(n)$, which grows only linearly in k, the latter is dominated as k grows, simplifying the worst-case complexity to $O(k \cdot T_{\mathcal{M}_\theta}(|Q|) + k^{a+1} \cdot c_{\max}^a)$.

Compared to Aerial+, the single-target rule learning paradigm, instantiated with TFMs, requires k forward passes per probe (one per feature), resulting in a factor of k slowdown during rule extraction. However, when using pretrained foundation models, it eliminates the training phase, which can be advantageous for small datasets or rapid deployment. Note that tree-based models, specifically RandomForest, are faster than Aerial+ despite the k factor slowdown, as shown in Figure 3.

B Experimental Details

This section includes extended experimental details that complement Section 4 and Appendix C which provides extended experimental results.

The experiments are run on a machine with an NVIDIA H100 GPU with 94 GiB memory, and a 4-core AMD EPYC 7F72 CPU per task.

Open-source repository. Source code of all the experiments, including reusable functions and the datasets, is openly available at https://github.com/DiTEC-project/tabprobe.

B.1 Datasets

10 datasets from the canonical UCI ML repository (Kelly et al., 2023) are used across Settings 1 and 2, summarized in Table 4. 5 small medical tabular datasets (∼100 rows) evaluate TFMs in low-data scenarios, while 5 larger datasets (300 to 8000 rows) emulate scenarios with sufficient data to train ML models. All datasets have a class label to evaluate downstream classification as part of interpretable ML models (CBA (Liu et al., 1998) and CORELS (Angelino et al., 2018)). Numerical columns are discretized using equal-frequency discretization with 10 bins. CSV versions of the datasets are available in the *data* folder of our repository.

Table 4: Datasets from the canonical UCI ML repository (Kelly et al., 2023) used in Settings 1 and 2.

Dataset	**Columns**	**Items**	**Rows**
Small Tabular Datasets			
Acute Inflammations	6	22	120
Hepatitis	19	89	155
Cervical Cancer	19	197	72
ASD Screening (Adolescent)	21	95	104
Fertility	9	36	100
Larger Tabular Datasets			
Breast Cancer	9	43	286
Congressional Voting Records	16	34	435
Mushroom	22	118	8124
Chess (KR vs. KP)	35	73	3196
Spambase	57	157	4601

B.2 Rule Quality Metrics

Table 5 defines the 5 rule quality metrics used in Setting 1. Zhang's metric (Yan et al., 2009) measures genuine association strength beyond statistical independence, controlling for marginal statistics by construction. X' in Zhang's metric formula refers to the negation, absence of X. Interestingness (Ghosh & Nath, 2004) measures how unexpected a rule is relative to its antecedent and consequent supports.

Table 5: Formulas and ranges for rule quality metrics for a rule $X \to Y$ over dataset D.

Metric	Formula	Range
Support	$\lvert\{t \in D : (X \cup Y) \subseteq t\}\rvert / \lvert D\rvert$	$[0, 1]$
Confidence	$\sup(X \to Y)/\sup(X)$	$[0, 1]$
Coverage	$\lvert\{t \mid \exists (X \to Y) \in \mathcal{R} : X \subseteq t\}\rvert / \lvert D\rvert$	$[0, 1]$
Zhang (Yan et al., 2009)	$\frac{\text{conf}(X \to Y) - \text{conf}(X' \to Y)}{\max(\text{conf}(X \to Y), \text{conf}(X' \to Y))}$	$[-1, 1]$
Interestingness (Ghosh & Nath, 2004)	$\frac{\sup(X \to Y)}{\sup(X)} \cdot \frac{\sup(X \to Y)}{\sup(Y)} \cdot (1 - \frac{\sup(X \to Y)}{\lvert D\rvert})$	$[0, 1]$

B.3 Reproducibility and Parameter Settings

Fixed execution seeds. All experiments are repeated with the following 10 fixed seeds listed in Table 6.

Table 6: Fixed execution seeds for reproducibility.

Seeds
42, 1608637542, 1273642419, 1935803228, 787846414, 996406378, 1201263687, 423734972, 415968276, 670094950

Parameter settings for all framework instantiations and baselines are listed in Table 7. FP-Growth is implemented with MLxtend version 0.23.3 (Raschka, 2018), Aerial+ with PyAerial version 1.0.24 (Karabulut et al., 2025c), the official xgboost Python package version 2.1.1[2] for XGBoost, version 1.1.13 of TabDPT from its original repository, version 2.1.1 for TabICL and version 8.0.2 for TabPFN.

Table 7: Parameter settings for framework instantiations and baselines across experimental settings. a is the antecedent length. $^{\dagger}\tau_a = 0.5$ for Setting 1, $\tau_a = 0.3$ for Setting 2 to obtain sufficient classification rules. ‡min_supp $\in \{0.5, 0.3, 0.2, 0.1\}$ for Setting 1, $\in \{0.3, 0.2, 0.1, 0.05, 0.01\}$ for Setting 2, as classification rules tend to have lower support.

Method	Parameters
TabPFNv3	$a = 2$, $\tau_a \in \{0.5, 0.3\}^{\dagger}$, $\tau_c = 0.8$, n_estimator $= 8$
TabICLv2	$a = 2$, $\tau_a \in \{0.5, 0.3\}^{\dagger}$, $\tau_c = 0.8$, n_estimator $= 8$
TabDPT	$a = 2$, $\tau_a \in \{0.5, 0.3\}^{\dagger}$, $\tau_c = 0.8$, n_estimator $= 8$
Aerial+	$a = 2$, $\tau_a \in \{0.5, 0.3\}^{\dagger}$, $\tau_c = 0.8$, grid search-tuned per dataset
XGBoost	$a = 2$, $\tau_a \in \{0.5, 0.3\}^{\dagger}$, $\tau_c = 0.8$, Optuna-tuned per dataset
FP-Growth	$a = 2$, min. confidence= 0.8, min. support $\in \{0.5, 0.3, 0.2, 0.1\}^{\ddagger}$

Dataset-specific parameter adjustments. In Setting 2, τ_a is further reduced for specific datasets while running CBA to obtain sufficient classification rules of lower support. For *cervical cancer behavior risk* and *breast cancer* datasets, $\tau_a = 0.1$ is used.

Aerial+ architecture parameters. Aerial+ utilizes an undercomplete denoising autoencoder whose architecture is optimized per dataset via grid search. The resulting parameters are listed in Table 8; all other parameters are kept at default values.

B.4 Model Calibration Diagnostics

Algorithms 1 and 2 interprets model outputs as conditional probability distributions. This section provides calibration diagnostics for every model instantiation showing at what level predicted model outputs reflect observed empirical probability distributions. We do this by providing expected calibration error (ECE) (Naeini et al., 2015), Brier scores (Glenn et al., 1950), and reliability diagrams (DeGroot & Fienberg, 1983).

[2] `https://github.com/dmlc/xgboost`

Table 8: Optimized Aerial+ architecture and training parameters per dataset.

Dataset	Batch Size	Layers	Epochs
Small Tabular Datasets			
Acute Inflammations	2	4	10
Hepatitis	2	4	10
Cervical Cancer	1	8	20
ASD Screen. (Adolescent)	2	4	10
Fertility	2	4	10
Larger Tabular Datasets			
Breast Cancer	2	4	5
Congressional Vot. Recs.	4	2	2
Mushroom	64	64	2
Chess (KR vs. KP)	8	[16, 4]	5
Spambase	64	64	2

ECE measures how well a model's predicted probabilities match the empirical frequency with which the corresponding outcome actually occurs. As an example, given a set of marked antecedents, if a model predicts 70% strength for a consequent item, does the empirical probability, confidence, match that 70% prediction? Predictions are partitioned into B (10) equal-width bins by predicted probability and compute:

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{n_b}{N} \left| \hat{p}_b - o_b \right|, \tag{11}$$

where n_b is the number of predictions in bin b, $N = \sum_b n_b$ is the total number of predictions, $\hat{p}_b$ is the mean predicted probability in bin b, and o_b is the observed frequency of the outcome in that bin.

The Brier score complements ECE by scoring predictions individually rather than in bins:

$$\mathrm{Brier} = \frac{1}{N} \sum_{i=1}^{N} \left(\hat{p}_i - y_i \right)^2, \tag{12}$$

where $y_i \in \{0, 1\}$ is the observed outcome probability for an instance i.

Reliability diagrams visualize the same per-bin quantities underlying ECE by plotting the mean predicted probability $\hat{p}_b$ against the observed frequency o_b for each bin. In reliability diagrams, a perfectly calibrated model falls on the diagonal $\hat{p}_b = o_b$. The points above the diagonal indicate under-confidence, and below the diagonal indicate over-confident predictions.

Reliability diagrams, ECE and Brier scores calculated over every model prediction, probes, on all datasets. The **results** per dataset and average over all datasets are shown in Table 9 and Figure 4. ECE of all methods except XGBoost was equal or under 3%, while XGBoost had an ECE of 6%. Brier scores of TFMs were all under 12%, while Aerial+ and random forest closely followed with 12.5% and 12.3% respectively. Brier score of XGBoost was 14.3%. The reliability diagrams, when averaged over all datasets, show that methods are closely following the diagonal curve. However, XGBoost showed significant deviations from the diagonal on acute inflammations, fertility, breast cancer, mushroom, and spambase datasets.

We interpret the results as that all methods except XGBoost were reasonably calibrated. We then calibrated XGBoost using Platt scaling (Platt et al., 1999) on small datasets, acute inflammations and fertility, and using isotonic calibration (Zadrozny & Elkan, 2002) on large datasets, breast cancer, mushroom and spambase. Diagnostic results for XGBoost after calibration improved on all datasets except the acute inflammations. The diagnostics after the calibration on the other 4 datasets were also included in Table 9 and Figure 4.

Table 9: Per-dataset and average ECE and Brier score for every model instantiation. Bold marks the best (lowest) value on average. Calibrated XGBoost (XGB cal.) is calibrated on fertility, breast cancer, mushroom and spambase datasets. TabPFNv3 has the lowest average ECE and Brier scores, while XGBoost has significantly higher scores than any other methods. XGB cal. has substantially improved over XGB in ECE and minimally in Brier score. (RF: Random Forest, XGB: XGBoost)

Dataset	**Aerial+**	**RF**	**TabDPT**	**TabICLv2**	**TabPFNv3**	**XGB**	**XGB (cal.)**
				Expected Calibration Error (ECE)			
Acute Inflammations	0.033	0.016	0.034	0.051	0.047	0.062	0.157
Hepatitis	0.010	0.018	0.012	0.018	0.012	0.028	-
Cervical Cancer	0.020	0.017	0.010	0.008	0.013	0.007	-
ASD Screening	0.017	0.011	0.019	0.018	0.011	0.014	-
Fertility	0.007	0.012	0.010	0.027	0.009	0.139	0.011
Breast Cancer	0.020	0.013	0.011	0.012	0.012	0.047	0.023
Congressional Voting	0.067	0.058	0.023	0.013	0.021	0.052	-
Mushroom	0.030	0.011	0.059	0.053	0.037	0.131	0.097
Chess (KR vs. KP)	0.029	0.048	0.032	0.029	0.019	0.058	-
Spambase	0.066	0.074	0.033	0.041	0.045	0.061	0.066
Average	0.030	0.028	0.024	0.027	**0.023**	0.060	0.042
				Brier Score			
Acute Inflammations	0.140	0.139	0.139	0.143	0.142	0.166	0.215
Hepatitis	0.115	0.108	0.107	0.107	0.105	0.111	-
Cervical Cancer	0.074	0.074	0.076	0.078	0.073	0.076	-
ASD Screening	0.107	0.102	0.106	0.105	0.102	0.103	-
Fertility	0.148	0.144	0.145	0.149	0.146	0.183	0.146
Breast Cancer	0.128	0.126	0.120	0.119	0.119	0.141	0.137
Congressional Voting	0.196	0.171	0.150	0.143	0.143	0.187	-
Mushroom	0.094	0.103	0.104	0.102	0.095	0.143	0.129
Chess (KR vs. KP)	0.136	0.155	0.136	0.137	0.134	0.171	-
Spambase	0.116	0.108	0.095	0.096	0.097	0.146	0.146
Average	0.125	0.123	0.118	0.118	**0.115**	0.143	0.137

Table 10: Rule quality metrics (%) for XGBoost before and after calibration on 4 datasets that require calibration according to diagnostics.

Dataset	**Method**	**#Rules**	**Support**	**Confidence**	**Zhang**	**Interest.**	**Coverage**
Fertility	XGBoost	279	20.5	67.9	12.0	21.3	100
	XGBoost (cal.)	41	39.7	86.3	-3.5	39.3	100
Breast Cancer	XGBoost	1	61.9	88.1	56.5	71.3	70.3
	XGBoost (cal.)	12	21.7	54.4	6.7	21.9	100
Mushroom	XGBoost	4326	19.8	62.0	8.7	21.4	100
	XGBoost (cal.)	1819	29.2	67.1	7.5	30.9	100
Spambase	XGBoost	27530	57.1	90.9	2.8	57.5	100
	XGBoost (cal.)	34324	55.0	90.5	3.7	55.4	100

We then compare the rule quality results on the 4 datasets where the calibration improved diagnostics. Table 10 shows that calibration changes rule quality substantially, however, not always for the better. On the Breast Cancer dataset, the uncalibrated model extracts a single high-support, high-confidence rule that covers 70.3% of the data. After calibration, the model extracts 12 rules covering the entire dataset, but at a lower confidence (54.4% vs. 88.1%) and a sharply reduced Zhang score (6.7% vs. 56.5%). This indicates that the additional rules are individually less discriminative and the previously overconfident single rule no longer dominates. On the Fertility dataset, calibration reduces the number of rules from 279 to 41 and raises confidence from 67.9% to 86.3%, but Zhang becomes negative (-3.5%), meaning the calibrated rules are, on

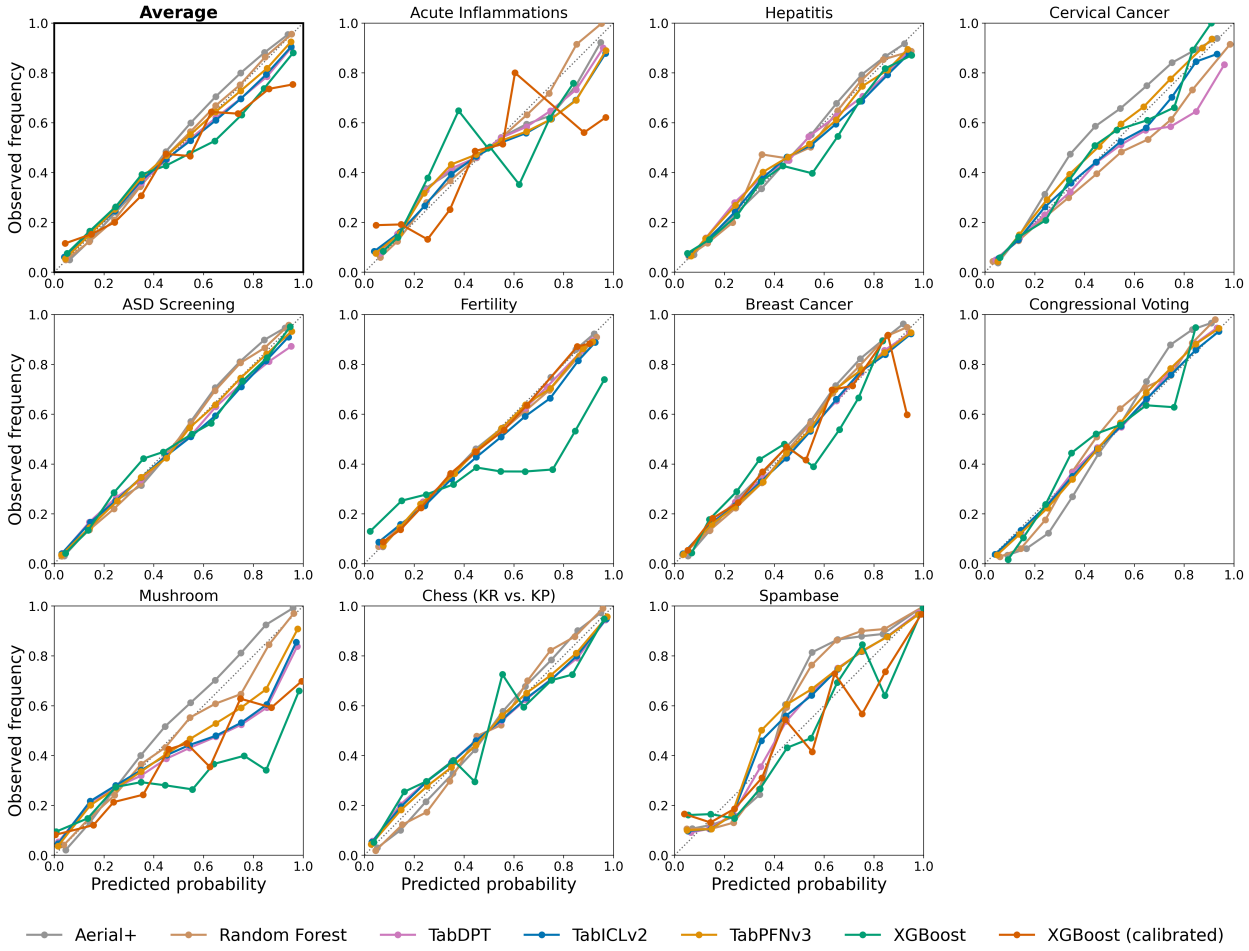


Figure 4: Reliability diagrams for all model instantiations per dataset and average over all the datasets. XGBoost is calibrated on the fertility and acute inflammations dataset using sigmoid calibration, and on the breast cancer, mushroom, and spambase datasets using isotonic calibration. On the acute inflammation dataset, XGBoost did not benefit from either calibration method. Calibrated XGBoost followed the diagonal line more closely than XGBoost. All other models followed the diagonal line relatively closely and were not calibrated when considered together with their ECE and Brier scores.

average, slightly anti-correlated with their consequents rather than genuinely associated. On the Mushroom dataset, calibration roughly halves the rule count while improving confidence and coverage with a small drop in Zhang score. On the Spambase dataset, the effect of calibration is mild. These results show that calibrating XGBoost's probability estimates does not necessarily improve the statistical rule quality.

The experimental results for XGBoost that are presented throughout the paper refer to XGBoost after calibration, since calibration diagnostics confirm these probability estimates better reflect empirical outcome frequencies, which is the property τ_a and τ_c are defined to threshold.

C Extended Experimental Results

This section presents extended experimental results for experimental setting 1 (Section 4.1) and setting 2 (Section 4.2), provides additional experiments and examples of learned rules.

C.1 Extended Rule Quality Evaluation

Table 11 shows rule quality evaluation results for small tabular datasets. Even small tabular data may lead to rule explosion, as FP-Growth produces 4088 rules on Autism Screening Adolescent with a 10% min support threshold, more than four times as many rules as any framework instantiation. All framework instantiations learn substantially more compact rule sets than FP-Growth, but their association strength (Zhang) varies across datasets. RF achieves the highest Zhang on Acute Inflammations and Cervical Cancer Behavior Risk, however, only by producing very few rules (3.3 to 3.9) at reduced data coverage (76.7 to 88.3%). TFMs lead on Fertility, where TabICLv2 reaches a Zhang of 11.1, and FP-Growth narrowly leads on Hepatitis, ahead of RF by only 0.1. XGBoost's performance is inconsistent, ranging from a negative Zhang on Fertility (-3.5) despite reasonable confidence, to the highest Zhang of any method on Autism Screening Adolescent (31.1). TFMs and Aerial+ are the only framework instantiations to maintain full or near-full coverage across all five datasets, with TFMs' coverage dropping only on Cervical Cancer Behavior Risk (87.3 to 98.5%). Aerial+ produces the lowest association strength on average across all methods, though it exceeds several baselines on individual datasets such as Fertility. We attribute this to known performance drops in neural networks with low data. TFM confidence scores are lower than the baselines on most datasets, with Fertility and Cervical Cancer as exceptions where TabPFNv3 and TabDPT are competitive with or exceed XGBoost and Aerial+. We argue, and experimentally show in Setting 2 (see Section 4.2), that this is due to TFM-learned rules capturing more generalizable patterns as opposed to dataset-specific idiosyncrasies. This is partially implied by higher association strength and interestingness scores in TFM rules compared to Aerial+, and a relatively higher number of rules than XGBoost and RF. The results in Setting 2 show that TFM-learned rules outperform the baselines when applied to unseen data in classification tasks.

Table 12 shows rule quality evaluation results for larger tabular datasets. On these datasets, the impact of rule explosion is more visible, e.g., as FP-Growth produces more than 100,000 rules on the Spambase dataset even with a 30% min support threshold. All framework instantiations learn substantially more compact rule sets than FP-Growth, and their association strength again varies across datasets. RF achieves the highest Zhang on Breast Cancer and Congressional Voting datasets, however, only by producing very few rules (1.8 to 5.3) at reduced data coverage (56.0 to 78.8%). TFMs lead on Chess and Spambase, while FP-Growth (0.01) leads on Mushroom by using roughly four times more rules than TFMs and XGBoost, 14 times more than RF, and 8 times more than Aerial+. XGBoost's performance is inconsistent, ranging from a competitive second-best Zhang on Congressional Voting (94.5) to the lowest confidence of any method on Breast Cancer (54.4). TFMs and Aerial+ are the only instantiations to maintain full or near-full coverage across all five datasets, while XGBoost's coverage drops sharply on Congressional Voting (75.9%) and RF's drops on both Breast Cancer and Congressional Voting.

Table 11: Rule quality evaluation of TFMs and baselines on small tabular datasets, with ± denoting SD across seeds for non-deterministic methods. TFMs maintain full or near-full coverage with far fewer rules than FP-Growth (FP-G) at low support thresholds, showing that rule explosion persists even in small data. TFMs match or exceed XGBoost's Zhang on most datasets while retaining full coverage. XGBoost's Zhang turns negative on Fertility despite reasonable confidence. RandomForest (RF) reaches the highest Zhang on most datasets but sacrifices coverage on Acute Inflammations and Cervical Cancer, where TFMs remain at or near full coverage. Across seed variation is higher than larger dataset, and is largest on Cervical Cancer.

Dataset	Algorithm	#Rules	Support	Confidence	Zhang	Interestingness	Coverage
Acute Inflammations	TabICL	60.4	31.1±0.9	78.2±2.4	56.6±6.9	45.0±2.0	100
	TabPFN	56.7	31.7±1.1	79.5±2.7	57.7±6.1	46.1±1.5	100
	TabDPT	52.5	32.4±0.8	80.8±2.1	60.8±6.1	47.5±2.1	100
	Aerial+	138	25.1±0.7	87.7±1.0	46.2±1.6	39.4±0.8	100
	RF	3.3	39.8±2.6	100	100	56.1±4.1	88.3±4.1
	XGB	16	25.9	86.4	59.7	34.2	100
	FP-G (0.5)	1	50.8	100	49.2	66.7	50.8
	FP-G (0.3)	46	37.7	92.4	62.1	58.8	100
	FP-G (0.2)	75	32.7	95.3	60.6	53.7	100
	FP-G (0.1)	123	25.6	96.1	55.1	42.6	100
Hepatitis	TabICL	673	42.5±2.1	82.7±1.5	17.2±3.6	44.5±2.5	100
	TabPFN	546	44.3±0.8	83.7±1.0	18.4±1.8	46.3±0.7	100
	TabDPT	566	42.1±2.7	82.1±1.8	15.2±4.0	44.0±3.0	100
	Aerial+	677	37.1±2.1	87.2±1.3	16.4±2.8	40.9±1.9	100
	RF	289	48.0±1.3	88.5±0.6	21.6±4.8	50.1±1.5	100
	XGB	513	41.6	82.2	17.1	43.9	100
	FP-G (0.5)	304	58.9	87.4	21.7	62.4	100
	FP-G (0.3)	835	45.9	88.7	19.4	49.7	100
	FP-G (0.2)	1153	40.2	88.9	18.0	43.8	100
	FP-G (0.1)	2340	26.5	89.9	18.3	29.6	100
Fertility	TabICL	47.5	36.3±2.1	85.8±2.9	11.1±10.4	36.7±1.9	100
	TabPFN	42.8	39.8±0.7	87.0±1.3	2.1±7.7	39.6±0.7	100
	TabDPT	42.2	39.7±0.7	87.7±1.3	7.5±8.8	39.8±0.4	100
	Aerial+	73.4	31.6±2.0	86.7±1.2	8.4±1.8	33.0±1.7	99.9±0.3
	RF	42.0	40.2±0.1	87.5±0.4	4.0±3.4	40.1±0.2	100
	XGB	41	39.7	86.3	-3.5	39.3	100
	FP-G (0.5)	10	59.6	88.9	7.9	60.1	100
	FP-G (0.3)	52	41.8	88.5	3.6	42.1	100
	FP-G (0.2)	109	32.8	88.5	5.9	33.7	100
	FP-G (0.1)	287	20.5	89.3	10.3	22.0	100
Cervical Cancer	TabICL	10.4	33.3±8.1	82.7±4.3	31.2±19.3	36.7±9.6	87.3±15.3
	TabPFN	11.2	43.1±4.7	94.0±2.4	56.4±12.1	46.5±5.1	89.2±5.4
	TabDPT	37.6	29.6±8.5	80.7±8.7	38.4±15.9	33.2±9.5	98.5±2.9
	Aerial+	24.1	29.6±6.3	86.2±6.8	28.3±10.2	35.4±5.1	92.9±9.1
	RF	3.9	51.7±5.7	97.1±0.9	75.7±7.9	56.0±5.9	76.7±4.6
	XGB	9	41.5	90.4	43.8	44.7	98.6
	FP-G (0.5)	2	61.1	96.8	22.8	65.9	80.6
	FP-G (0.3)	27	40.4	90.7	18.4	43.9	98.6
	FP-G (0.2)	83	29.2	92.0	24.3	34.3	100
	FP-G (0.1)	476	16.6	92.7	31.6	23.8	100
Autism Screening	TabICL	862	38.9±3.7	84.9±1.6	20.2±2.1	40.9±4.0	100
	TabPFN	714	43.9±1.9	88.1±1.0	17.5±3.7	45.6±2.0	100
	TabDPT	773	39.0±3.7	83.1±2.9	9.1±6.8	40.1±4.1	100
	Aerial+	664	33.8±3.2	89.9±2.8	14.2±5.1	37.0±3.9	100
	RF	488	45.6±1.4	90.8±0.8	12.7±7.1	46.0±1.5	100
	XGB	645	47.4	89.6	31.1	49.4	100
	FP-G (0.5)	445	58.5	89.0	17.6	61.0	100
	FP-G (0.3)	1193	46.3	88.9	14.8	48.8	100
	FP-G (0.2)	1982	37.5	89.5	13.5	39.8	100
	FP-G (0.1)	4088	25.5	90.2	14.8	28.0	100

Table 12: Rule quality evaluation of all methods on larger tabular datasets, with ± denoting SD across seeds. TFMs maintain near-full coverage with far fewer rules than FP-Growth (FP-G), which needs an order of magnitude more rules to match coverage, illustrating rule explosion. XGBoost's confidence and Zhang trail TFMs on nearly every dataset. RandomForest (RF) reaches high Zhang or confidence only at the cost of coverage or rule count (and shows negative Zhang despite high confidence on Mushroom). TFMs sustain competitive Zhang alongside full coverage. Variation across seeds is smaller than small tabular datasets.

Dataset	Algorithm	#Rules	Support	Confidence	Zhang	Interestingness	Coverage
Breast Cancer	TabICL	35.1	45.1±2.4	86.7±1.3	52.3±5.4	52.3±3.1	96.3±2.4
	TabPFN	26.8	48.5±2.2	88.5±1.2	59.9±3.6	57.0±2.9	92.3±3.6
	TabDPT	27.0	47.5±2.0	88.2±1.5	58.5±5.3	55.8±2.8	91.7±3.5
	Aerial+	15.7	43.9±4.5	88.1±4.6	36.4±6.9	52.4±4.8	90.2±4.8
	RF	5.3	58.1±2.5	93.6±0.8	78.1±2.8	71.4±3.0	78.8±1.9
	XGB	12	21.7	54.4	6.7	21.9	100
	FP-G (0.5)	22	59.5	87.6	46.5	68.7	94.4
	FP-G (0.3)	51	44.4	87.8	32.1	51.4	94.4
	FP-G (0.2)	112	33.4	87.7	26.7	39.2	97.9
	FP-G (0.1)	208	24.5	87.3	23.4	29.1	100
Congressional Voting	TabICL	566	33.4±1.0	88.5±1.9	83.2±5.0	56.0±2.7	99.8±0.1
	TabPFN	423	34.6±0.6	90.1±1.1	85.8±2.3	58.1±1.2	99.8±0.1
	TabDPT	313	34.5±1.1	89.4±2.0	83.1±4.9	56.3±2.8	99.8±0.1
	Aerial+	224	34.8±1.9	92.0±2.0	61.4±3.5	56.6±3.2	97.8±5.6
	RF	1.8	49.2±1.3	99.7±0.5	99.6±0.8	79.9±2.5	56.0±12.1
	XGB	11	28.9	92.8	94.5	67.0	75.9
	FP-G (0.5)	14	51.6	89.8	84.7	78.4	57.6
	FP-G (0.3)	1016	35.5	89.1	70.6	60.6	100
	FP-G (0.2)	2024	30.7	88.6	65.0	52.9	100
	FP-G (0.1)	2867	26.0	88.5	60.5	44.7	100
Mushroom	TabICL	1913	29.0±1.2	73.9±1.9	21.6±2.0	32.6±1.3	100
	TabPFN	1715	29.3±0.9	76.4±1.6	25.2±4.1	34.0±1.2	100
	TabDPT	1752	27.1±1.4	67.3±2.7	11.8±4.7	30.3±1.6	100
	Aerial+	858	37.2±1.7	95.5±1.1	8.5±1.5	39.4±1.5	100
	RF	488	45.1±0.7	91.3±0.5	-5.2±2.6	45.1±0.8	100
	XGB	1819	29.2	67.1	7.5	30.9	100
	FP-G (0.5)	245	65.8	94.5	6.8	66.0	100
	FP-G (0.3)	1247	44.3	93.7	17.8	48.6	100
	FP-G (0.2)	3147	31.8	95.0	24.5	38.7	100
	FP-G (0.1)	6882	22.2	95.3	27.6	29.3	100
Chess (KR vs. KP)	TabICL	13710	56.4±0.5	90.7±0.5	10.0±1.4	57.2±0.6	100
	TabPFN	12989	57.6±0.6	91.9±0.8	10.8±1.5	58.4±0.7	100
	TabDPT	14454	56.4±0.5	90.0±0.4	8.7±1.6	57.0±0.5	100
	Aerial+	11902	57.4±1.7	94.8±0.8	1.3±0.3	57.6±1.7	100
	RF	9648	56.2±1.3	92.8±0.9	6.8±1.3	56.5±1.3	100
	XGB	11688	49.3	84.4	8.2	49.3	100
	FP-G (0.5)	12752	69.7	93.0	4.9	69.0	100
	FP-G (0.3)	21064	57.4	93.1	4.5	57.2	100
	FP-G (0.2)	27604	49.8	93.3	4.8	49.9	100
	FP-G (0.1)	35358	42.1	93.5	5.6	42.6	100
Spambase	TabICL	46180	77.3±0.6	94.8±0.2	5.2±0.3	77.7±0.6	100
	TabPFN	44269	78.3±0.4	94.8±0.2	5.1±0.2	78.7±0.4	100
	TabDPT	9341	63.6±1.0	90.1±0.6	20.0±2.4	65.2±1.0	100
	Aerial+	23075	81.4±1.6	97.7±1.3	2.5±1.0	81.7±1.5	100
	RF	30490	80.9±0.3	97.0±0.1	2.7±0.3	81.0±0.3	100
	XGB	34324	55.0	90.5	3.7	55.4	100
	FP-G (0.5)	96094	77.8	92.2	4.2	76.5	100
	FP-G (0.3)	109312	73.6	92.2	4.7	72.6	100
	FP-G (0.2)	113168	72.0	92.3	4.9	71.2	100
	FP-G (0.1)	131567	63.4	93.2	4.9	62.6	100

C.2 Extended Downstream Task Evaluation

This section presents extended downstream task evaluation results per dataset (see Section 4.2).

Experimental setup for downstream task evaluation is shown in Figure 5. CBA Liu et al. (1998) and CORELS Angelino et al. (2018) are two well-known interpretable ML models used in high-stakes decision-making. CBA receives a set of association rules, while CORELS receives a set of frequent itemsets to select and order a subset to build a rule-based classifier. Each tabular dataset is first split into training and test folds for 5-fold cross-validation. All methods are run on the training folds to learn association rules for CBA and frequent itemsets for CORELS (see Section E for an adaptation of Algorithm 2 to frequent itemset learning). Rule-based classifiers are then built using CBA and CORELS and evaluated on the test folds based on accuracy, precision, recall, and F1 score.

The results of classification tasks per dataset are given in Figures 6 and 7. No single method uniformly dominates across classifiers and dataset sizes. By looking at the F1 scores with CORELS on small tabular datasets, we see that the best performing TFM in each dataset are ahead of Aerial+ and of tree-based models on 3 out of 5 datasets, and comparable on 2 datasets. With comparable, we are referring to being $\pm 1\%$ within. Compared to FP-Growth variations, TFMs are ahead on 2, have comparable results on 2, and have 2% worse results on the Cervical Cancer dataset compared to FP-G (0.2). On large datasets, TFMs are comparable to tree-based models on all 5 datasets. Against Aerial+, they are ahead on 2, Breast Cancer and Spambase, and have comparable results on the remaining 3 datasets. Lastly, compared to FP-Growth variations on larger datasets, the best-performing TFM is always comparable to the best-performing FP-Growth variation, despite using a substantially lower number of rules as shown in Setting 1.

CBA only considers support and confidence of the rules when building classifiers and not association strength or any other generalizability metric. Random Forest did not result in enough classification rules in 3 of the 5 small datasets and 2 of the larger datasets, while XGBoost failed to find classification rules on the 2 of the small datasets. Note that this is despite lowering the $\tau_a = 0.1$ on those datasets. With CBA on small datasets, TFMs are behind tree-based models only on the Hepatitis dataset against XGBoost. Compared to Aerial+, they are comparable on 2, ahead on 2, and behind only on the Cervical Cancer dataset. Against FP-Growth, they are 4% behind on the Cervical Cancer dataset, 3% on Autism Screening and comparable in the remaining datasets. On large datasets with CBA, TFMs are ahead of tree-based models on all 5 datasets. Against Aerial+, they are ahead on 2, and have comparable results on the remaining 3. And finally, against FP-Growth, they are ahead on 1, comparable on 3, and behind on 1 dataset.

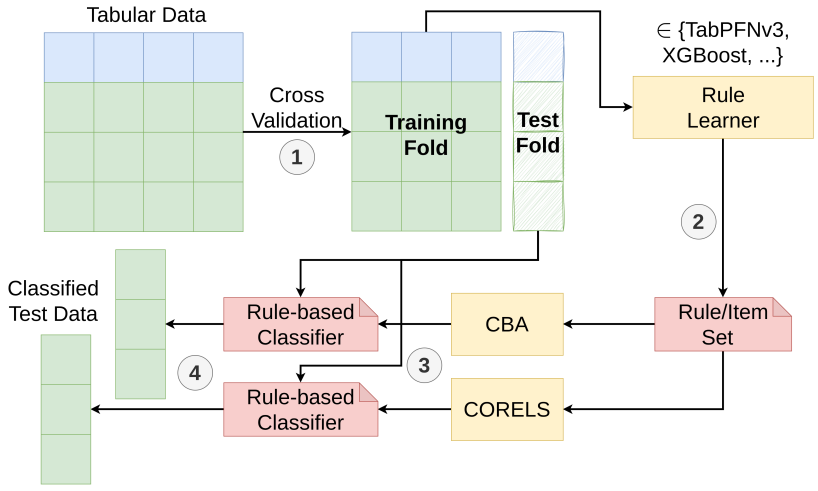


Figure 5: Experimental Setting 2 for downstream task evaluation. (1) Tabular data is split into training and test folds for cross-validation, (2) TFMs and baselines learn rules (CBA) or frequent itemsets (CORELS) on the training fold, (3) learned rules and itemsets are passed to CBA and CORELS to build a rule-based classifier, (4) classifiers are evaluated on the test fold. This process is repeated for 5-fold across 10 seeds.

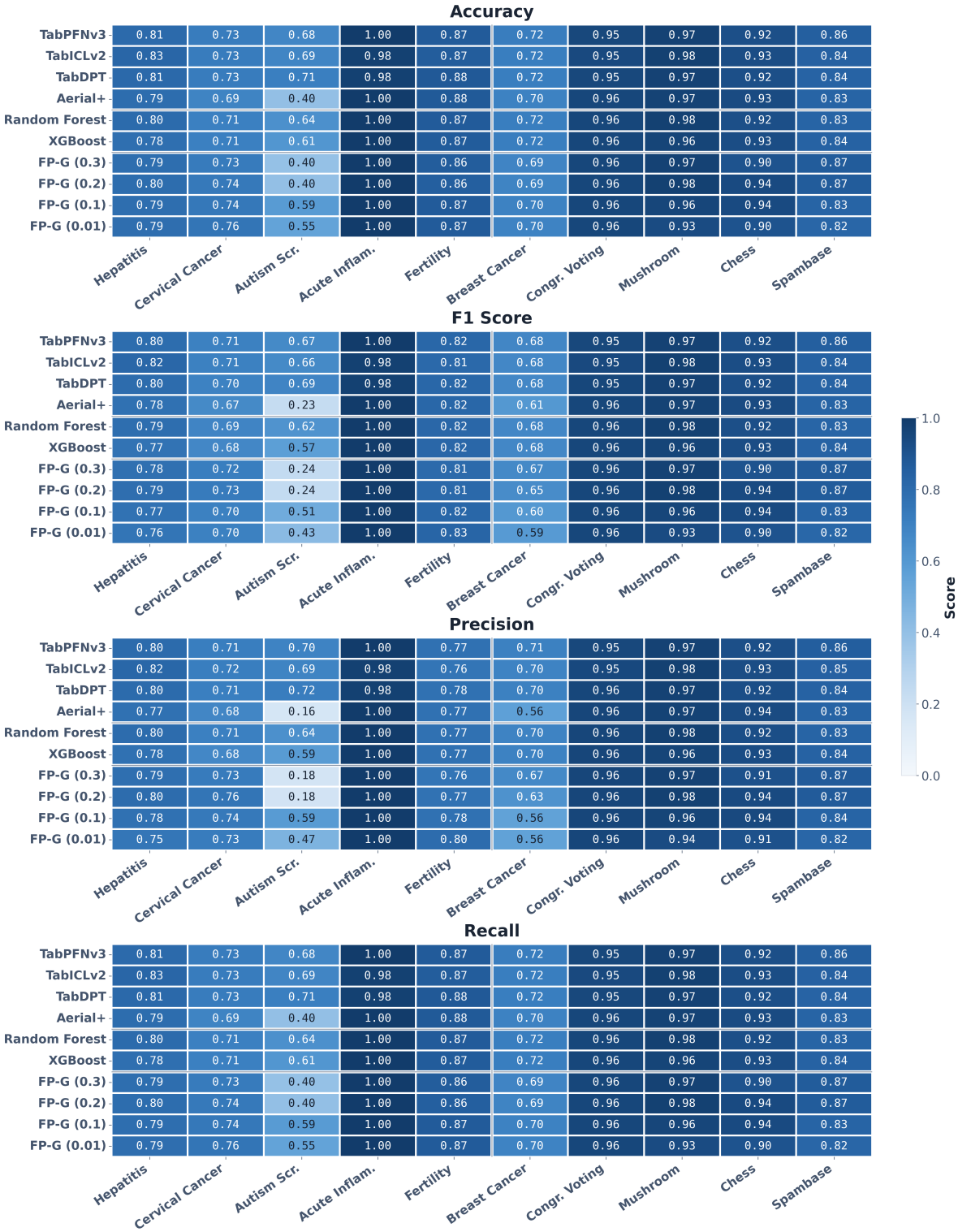


Figure 6: Classification results with CORELS on both small and larger tabular datasets. On small datasets, most methods perform comparably on Hepatitis, Acute Inflammations, and Fertility, while Aerial+ and high-support FP-Growth settings are behind the other methods on Cervical Cancer and, more substantially (∼30 percentage points), on Autism Screening. On larger datasets, all methods perform comparably (within ∼1%) except on Breast Cancer and Spambase, where Aerial+ trails the best-performing method by ∼2% and ∼3.5%, respectively.

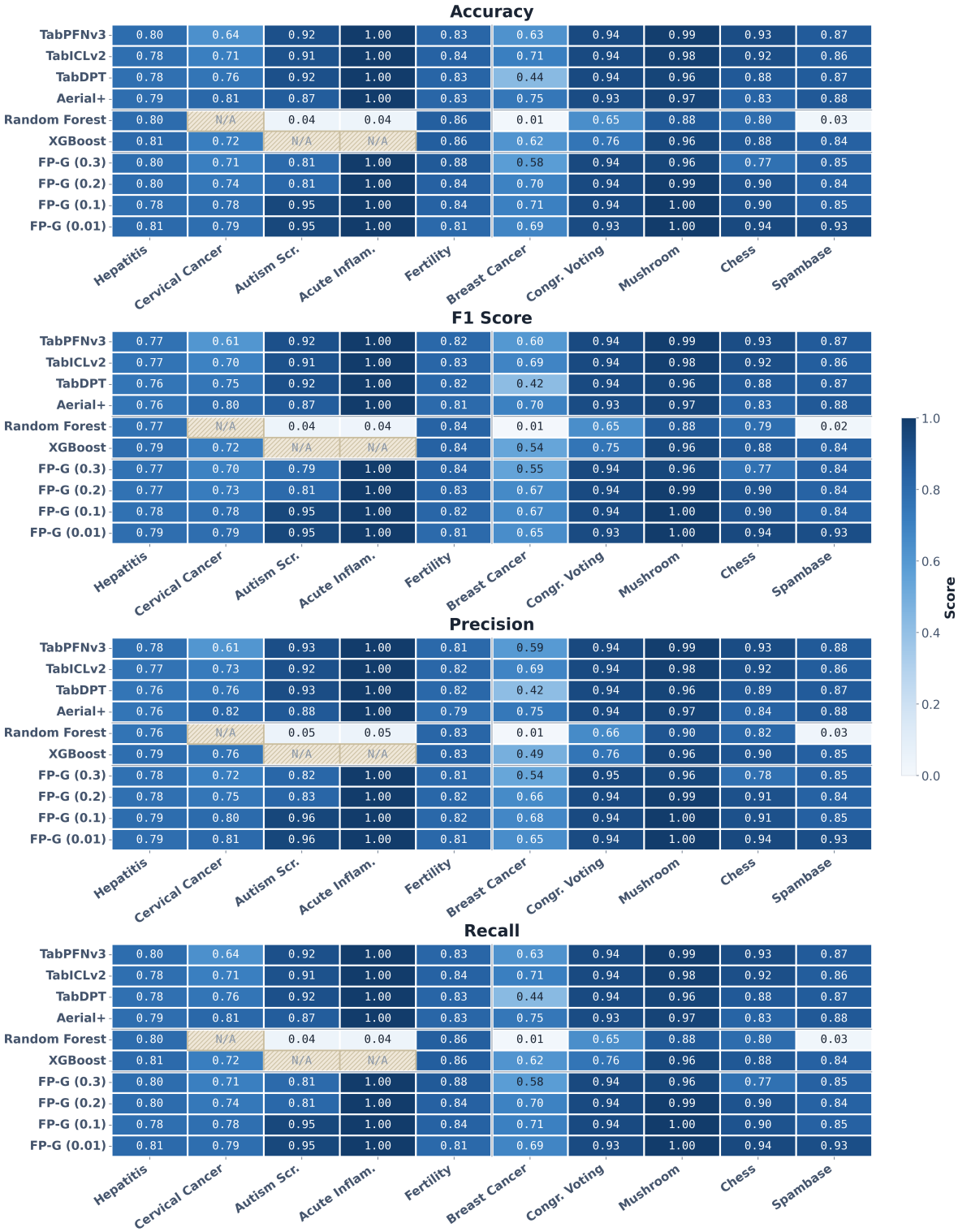


Figure 7: Classification results with CBA on all tabular datasets. With CBA, Random Forest could not find enough classification rules on 3 small and 2 larger datasets, while XGBoost could not find enough classification rules on 2 small datasets only. The best-performing TFM in each dataset performed better or comparably to other methods in 7 out of the 10 datasets. FP-Growth benefited significantly from lower min support thresholds. Note that CBA prioritizes rules of higher support and confidence rather than association strength and generalizability.

C.3 Binning sensitivity analysis

The experiments use 10 datasets, of which 6 contain numerical features: Acute Inflammations, Hepatitis, Cervical Cancer Behavior Risk, Autism Screening, Fertility and Spambase. As stated in Section B.1, numerical features are discretized into 10 bins using equal-frequency discretization. However, the number of bins can influence the number of rules and their statistical quality. Therefore, in this section, we repeat the rule quality experiments from Section 4.1 with different numbers of bins (3 and 5) and present the results. All the remaining parameters of each method are the same as in the main rule quality experiments.

Table 13 shows rule quality evaluation results for all methods as the number of bins changes. The number of rules decreases across most methods as the number of bins increases, with the exception of TapDPT, which has significantly more rules with 10 bins. Aerial+ shows the sharpest drop in rule count as the bins increase. The data coverage percentages remain largely the same for all methods, within less than 1% change across different numbers of bins. The confidence levels show a slight decrease in all FP-Growth variations, while Aerial+, TabPFNv3, and TabICLv2 remain around the same confidence levels as the number of bins increases. TabPDT has increased confidence by up to 1.3%, RandomForest showed non-monotonicity in confidence levels, and XGBoost's confidence dropped by 4.1% with larger bins. Zhang score is not uniformly affected by the bin counts, similar to the confidence levels. TabPFNv3 and RandomForest both have increased Zhang scores with more bins, TabDPT shows the opposite trend, and XGBoost is non-monotonic, dropping at 5 bins before rising again at 10. FP-Growth's metrics change only modestly across different bins, within $1-2\%$. These results suggest that the number of bins is a relevant hyperparameter for rule quality, its effect varies by model class, and that there is no single bin count uniformly optimal across all metrics or models.

Table 13: Average rule quality metrics (%) across 3, 5, and 10 equal-frequency bins, computed over the 6 datasets that contain numerical (and categorical) features: Acute Inflammations, Autism Screening Adolescent, Cervical Cancer, Fertility, Hepatitis, Spambase. Coverage remains stable across bin counts for most methods, with RandomForest showing a small decline as the bins increase. Rule count decreases with more bins for most methods, with the exception of TabPDT, whose rule count grows. Zhang scores increase with bin count in TabPFNv3, Aerial+, RandomForest, and most FP-Growth configurations, while it decreases in TabDPT and XGBoost's Zhang is non-monotonic.

Algorithm	**#Rules**	**Support**	**Confidence**	**Zhang**	**Interestingness**	**Coverage**
			3 Bins			
TabICLv2	11776	44.2	85.1	22.1	47.6	97.9
TabPFNv3	11680	48.1	88.4	24.5	51.5	98.1
TabDPT	269	34.9	82.8	32.1	40.3	99.7
Aerial+	16894	37.1	89.1	17.9	41.5	98.8
XGBoost	12585	44.2	88.7	24.3	46.5	99.8
RandomForest	11405	52.1	92.8	32.4	55.9	94.7
FP-G (0.5)	20628	63.5	93.6	19.4	67.9	88.6
FP-G (0.3)	31485	47.2	91.6	19.8	52.6	99.8
FP-G (0.2)	32725	39.0	92.5	19.9	44.4	100.0
FP-G (0.1)	34565	30.2	93.0	21.0	35.2	100.0
			5 Bins			
TabICLv2	9536	43.6	85.1	23.7	47.3	97.9
TabPFNv3	9392	48.1	88.6	25.7	51.6	98.1
TabDPT	357	37.8	83.1	26.2	41.9	99.7
Aerial+	8998	39.7	89.2	18.5	44.3	98.8
XGBoost	8183	46.3	86.0	17.7	48.1	99.8
RandomForest	8315	52.2	93.6	35.1	55.8	94.4
FP-G (0.5)	20025	62.3	92.9	20.1	66.8	88.6
FP-G (0.3)	21719	49.0	90.8	20.0	54.4	99.8
FP-G (0.2)	25547	40.4	91.8	20.7	46.0	100.0
FP-G (0.1)	42197	26.4	92.5	20.9	31.2	100.0
			10 Bins (Main Paper Setting)			
TabICLv2	7972	43.2	84.9	23.5	46.9	97.9
TabPFNv3	7607	46.8	87.9	26.2	50.5	98.2
TabDPT	1803	41.1	84.1	25.1	45.0	99.7
Aerial+	4109	39.7	89.2	19.3	44.6	98.8
XGBoost	4832	39.0	84.6	27.8	41.8	99.8
RandomForest	5219	51.0	93.5	36.1	54.9	94.2
FP-G (0.5)	16143	61.1	92.4	20.6	65.4	88.6
FP-G (0.3)	18578	47.6	90.2	20.5	52.7	99.8
FP-G (0.2)	19428	40.7	91.1	21.2	46.1	100.0
FP-G (0.1)	23147	29.7	91.9	22.5	34.8	100.0

C.4 Comparison to Maximal Itemset Mining

The experiments so far use 7 competitive baselines, including a SOTA neurosymbolic rule learner, Aerial+, and a canonical rule miner, FP-Growth. We now compare against another canonical baseline, FP-Max (Grahne & Zhu, 2003b;a), a maximal itemset mining method that is built on FP-Tree structures from FP-Growth. A maximal frequent itemset is an itemset whose size can not be increased by adding another item without losing the user-defined support threshold. FP-Max is known to address rule explosion in ARM. We repeat the rule quality experiments from Setting 1 with FP-Max using different minimum support thresholds, similar to FP-Growth, while keeping the remaining hyperparameters the same. FP-Max is implemented using MLxtend Raschka (2018).

Table 14: Average rule quality (%) of FP-Max. FP-Max could not find any rules in the Spambase dataset. Rule count in FP-Max is not monotonic in the support threshold, because the maximal-itemset filter drops a shorter itemset once a longer frequent superset is found. Across all the remaining datasets, FP-Max mined substantially fewer rules than FP-Growth, addressing rule explosion. However, in many cases, FP-Max was not able to reach full data coverage and led to lower Zhang values than other baselines given in Tables 11 and 12, specifically RandomForest and all 3 TFMs.

Algorithm	**#Rules**	**Supp.**	**Conf.**	**Zhang**	**Int.**	**Cov.**
Acute Inflammations						
FP-Max(0.5)	1	50.8	100.0	49.2	66.7	50.8
FP-Max(0.3)	13	36.4	93.3	69.5	60.0	100.0
FP-Max(0.2)	12	27.1	97.0	56.0	46.2	83.3
FP-Max(0.1)	5	11.0	93.6	32.4	15.5	48.3
Hepatitis						
FP-Max(0.5)	32	52.0	85.2	12.3	54.4	100.0
FP-Max(0.3)	45	32.0	90.9	14.7	35.0	99.4
FP-Max(0.2)	31	20.2	92.2	17.5	23.9	93.5
FP-Max(0.1)	53	10.4	92.8	16.3	12.0	91.6
Fertility						
FP-Max(0.5)	6	53.8	88.4	7.5	54.1	96.0
FP-Max(0.3)	27	35.5	88.2	3.2	35.6	100.0
FP-Max(0.2)	10	21.2	88.3	8.8	22.9	89.0
FP-Max(0.1)	46	10.7	89.2	11.4	12.0	97.0
Cervical Cancer Behavior Risk						
FP-Max(0.5)	2	61.1	96.8	22.8	65.9	80.6
FP-Max(0.3)	10	33.5	91.9	11.3	36.2	94.4
FP-Max(0.2)	23	22.5	93.4	17.5	25.7	94.4
FP-Max(0.1)	55	11.9	93.3	27.4	18.3	100.0
Autism Screening Adolescent						
FP-Max(0.5)	55	50.6	88.1	25.4	54.7	99.0
FP-Max(0.3)	39	31.2	90.5	6.0	32.4	100.0
FP-Max(0.2)	66	20.6	89.5	5.9	21.4	98.1
FP-Max(0.1)	64	10.6	91.1	7.3	11.8	86.5

Algorithm	**#Rules**	**Supp.**	**Conf.**	**Zhang**	**Int.**	**Cov.**
Breast Cancer						
FP-Max(0.5)	0	–	–	–	–	0.0
FP-Max(0.3)	14	31.0	88.1	19.9	35.7	88.5
FP-Max(0.2)	13	21.5	86.1	21.9	26.3	81.8
FP-Max(0.1)	10	10.9	85.8	22.5	14.2	74.8
Congressional Voting Records						
FP-Max(0.5)	8	50.3	89.1	84.2	75.8	69.4
FP-Max(0.3)	100	30.8	89.0	64.2	53.5	99.8
FP-Max(0.2)	49	20.8	87.2	46.6	32.5	98.9
FP-Max(0.1)	18	10.8	88.8	37.8	16.4	52.4
Mushroom						
FP-Max(0.5)	9	51.3	88.7	29.3	54.9	76.9
FP-Max(0.3)	3	31.3	100.0	0.0	31.3	54.5
FP-Max(0.2)	6	20.6	100.0	0.0	20.6	69.4
FP-Max(0.1)	3	10.1	100.0	0.0	10.1	20.3
Chess (KR vs. KP)						
FP-Max(0.5)	8	50.0	96.2	-2.2	49.0	87.7
FP-Max(0.3)	2	30.0	88.6	0.5	30.1	38.0
FP-Max(0.2)	0	–	–	–	–	0.0
FP-Max(0.1)	0	–	–	–	–	0.0
Averages across all datasets						
FP-Max(0.5)	12.1	52.5	91.6	28.6	59.4	66.0
FP-Max(0.3)	25.3	32.4	91.2	21.0	38.9	77.4
FP-Max(0.2)	21.0	21.8	91.7	21.8	27.4	70.8
FP-Max(0.1)	25.4	10.8	91.8	19.4	13.8	57.1

The results are given in Table 14. FP-Max was not able to find any rules in the Spambase dataset across all four minimum support thresholds. It was able to address rule explosion, as its rule count is substantially lower than FP-Growth (see Tables 11 and 12). Note that the number of rules in FP-Max is not monotonic based on the minimum support threshold, as the maximal-itemset filter in FP-Max drops a shorter itemset once a longer superset is found. FP-Max reached near-full data coverage in 6 out of 10 datasets with at least one of its minimum support threshold configurations. However, its average rule quality is far below any of the earlier baselines. The confidence levels are comparable to FP-Growth and higher than TFMs in most cases, while its Zhang score is significantly below all baselines, with the exception of FP-Max(0.5) which is higher than XGBoost and Aerial+. Overall, FP-Max can address rule explosion; however, at the cost of significant data coverage and reliability across datasets and configurations.

C.5 Statistical comparison of rule-mining methods

We test whether the rule-mining methods behave differently across datasets. The analysis includes the five small and five larger datasets, and the following ten methods: TabICLv2, TabPFN, TabDPT, Aerial+, XGBoost, RandomForest, and FP-Growth with minimum support values of 0.5, 0.3, 0.2, and 0.1. We analyze six metrics separately: number of rules, mean support, mean confidence, mean Zhang's metric, mean interestingness, and data coverage. Results are first averaged over repeated seeds within each dataset. We use datasets as the paired units because our claims concern generalization across datasets.

For each metric, we rank the ten methods within every dataset. Rank 1 denotes the highest observed value. We note that this convention describes direction only, as a higher value is not necessarily preferable for every metric, particularly the number of rules. Ties receive their average rank.

We first apply the **Friedman test** to each metric. Its null hypothesis is that all ten methods have the same rank distribution across the ten datasets. The test uses the within-dataset ranks and includes the standard correction for ties. We report the Friedman statistic as $\chi^2_F(9)$, where 9 is the number of degrees of freedom (the number of methods minus one), with a larger statistic meaning that the observed method ranks depart more strongly from the equal-ranks hypothesis; as well as the associated p-values.

When the Friedman test rejects its null hypothesis at the significance level $\alpha = 0.05$, we apply a two-sided **Nemenyi post-hoc test** to all method pairs. The Nemenyi test asks whether the average ranks of two methods differ by more than would be expected when all methods are compared together. For ten methods and ten datasets, the resulting critical difference is 4.284: a pair is significant when its average ranks differ by more than this value. We use the 0.05 significance level separately for each metric because each metric represents a distinct empirical question.

C.5.1 Results

As shown in Table 15, the Friedman test rejects the null hypothesis of equal method ranks for every metric. The evidence against equal ranks is strongest for the number of rules, mean support, and mean interestingness. It is weaker, but still statistically significant, for mean Zhang's metric and data coverage.

Table 15: Friedman and post-hoc results across ten datasets and ten methods. The Friedman statistic $\chi^2_F(9)$ tests whether all methods have equal ranks; 9 is its number of degrees of freedom, and larger values provide stronger evidence against equal ranks. The p-value is the probability of a result at least this extreme under the equal-ranks hypothesis. "Nemenyi pairs" counts the method pairs whose average ranks differ significantly at the 0.05 level.

Metric	$\chi^2_F(9)$	p-value	Nemenyi pairs
Number of rules	62.836	3.80×10^{-10}	12
Mean support	57.055	4.93×10^{-9}	10
Mean confidence	34.303	7.90×10^{-5}	5
Mean Zhang's metric	17.695	0.0389	1
Mean interestingness	52.625	3.44×10^{-8}	10
Data coverage	24.240	0.0039	0

Table 16 reports the average ranks and therefore shows the direction and size of the differences. Any pair whose ranks differ by more than 4.284 is significant under the Nemenyi test. The biggest differences are concentrated in the number of rules, mean support, and mean interestingness. For example, FP-Growth with minimum support 0.1 has a rule-count rank significantly closer to 1 than TabPFN, TabDPT, Aerial+, FP-Growth with minimum support 0.5, RandomForest, and XGBoost. FP-Growth with minimum support 0.5 has a support rank significantly closer to 1 than TabICLv2, TabDPT, Aerial+, FP-Growth with minimum support 0.2, FP-Growth with minimum support 0.1, and XGBoost. For mean confidence, five pairs are significant, with RandomForest ranking significantly ahead of TabICLv2, TabDPT, and XGBoost; for mean Zhang's metric, only TabPFN and Aerial+ differ significantly. Although the Friedman result for data coverage is significant, no individual Nemenyi comparison exceeds the critical difference. This can occur

Table 16: Average method ranks across the ten datasets. Rank 1 denotes the highest value. Differences larger than the Nemenyi critical difference of 4.284 are significant at $\alpha = 0.05$. Bold marks the method with the highest observed values (the lowest numerical rank) for each metric.

Method	Rules	Support	Confidence	Zhang	Interestingness	Coverage
TabICLv2	4.80	6.10	8.50	4.20	6.10	5.25
TabPFN	6.00	4.20	5.70	**3.70**	4.30	5.45
TabDPT	5.80	6.20	8.00	5.20	6.60	5.50
Aerial+	6.10	6.50	4.10	8.10	6.20	6.30
FP-Growth (0.5)	8.50	**1.40**	4.65	6.00	**1.70**	6.50
FP-Growth (0.3)	4.10	3.80	5.40	6.40	3.90	4.70
FP-Growth (0.2)	2.30	7.30	4.60	6.60	7.30	4.25
FP-Growth (0.1)	**1.10**	9.60	3.80	5.80	9.50	**4.15**
XGBoost	7.20	7.00	7.30	4.30	6.60	5.85
RandomForest	9.10	2.90	**2.95**	4.70	2.80	7.05

because the Friedman test combines evidence across all methods, whereas each post-hoc comparison must meet a stricter pairwise threshold.

C.6 Rule Quality at Larger Antecedent Length

In this section, we evaluate the rule quality at antecedent length $a = 3$ to complement the main results at length $a = 2$ given in Table 2 with a longer antecedent to control for the effect of rule list size. The experimental setup is the same as in the main rule quality experiments in Section 4.1. The results are given in Table 17. Similar to the results in Table 2, RandomForest and TFMs, specifically TabPFNv3 on small and all TFMS on larger datasets, maintain higher association strength than all other methods while producing far fewer rules than FP-Growth. The gap in rule count widens at $a = 3$. FP-Growth at minimum support threshold produces 818,765 rules on larger datasets, and TabDPT produces only 43,428 rules, 19x fewer rules with practically full coverage, while the gap was 7x at $a = 2$. Relatively lower confidence of TFMs is maintained at $a = 3$, as well as lower data coverage of RandomForest, similar to $a = 2$. However, as shown in Section 4.2, relatively lower confidence rules of TFMs are not necessarily bad, as they have led to higher predictive performance under cross-validation than FP-Growth, demonstrating that TFMs-learned rules are more generalizable to unseen data. These results confirm that the conclusions drawn in Section 4.1 are not specific to $a = 2$, and carry to longer antecedent lengths.

Table 17: Average rule quality metrics (%) with antecedent length $a = 3$, with $\pm$ denoting SD across seeds for non-deterministic methods. The results follow the same trend as in $a = 2$ given in Table 2: Random Forest and all 3 TFMs have led to higher association strength, with TabPFNv3 leading within TFMs; the gap in the number of rules is widened to 19x (TabDPT vs FP-G (0.1) on larger datasets) which was 7x at $a = 2$; TFMs maintained relatively lower confidence, while Random Forest had the lowest data coverage.

Datasets	Algorithm	#Rules	Support	Confidence	Zhang	Interestingness	Coverage
Small Tabular Datasets	TabICLv2	869	33.7 ± 3.0	83.1 ± 2.6	30.3 ± 8.5	38.5 ± 3.7	97.5 ± 3.1
	TabPFNv3	758	36.5 ± 2.1	87.1 ± 1.5	35.0 ± 7.0	41.3 ± 2.4	97.8 ± 1.1
	TabDPT	797	33.1 ± 3.4	83.4 ± 2.9	29.7 ± 8.3	37.8 ± 3.9	99.7 ± 0.6
	Aerial+	2008	21.6 ± 2.4	87.1 ± 2.7	21.8 ± 4.3	26.6 ± 2.4	98.6 ± 1.8
	RandomForest	615	39.9 ± 2.5	93.1 ± 0.7	46.7 ± 5.3	44.6 ± 3.0	93.0 ± 1.7
	XGBoost	1033	32.5	88.2	35.6	35.4	99.7
	FP-G (0.5)	334	56.7	92.5	25.4	62.0	86.3
	FP-G (0.3)	1734	40.3	90.1	26.8	47.0	99.7
	FP-G (0.2)	3499	30.8	91.3	28.0	37.6	100.0
	FP-G (0.1)	8976	19.5	92.3	30.2	25.8	100.0
Larger Tabular Datasets	TabICLv2	164123	43.0 ± 1.2	87.4 ± 0.9	37.4 ± 2.4	49.6 ± 1.5	99.2 ± 0.5
	TabPFNv3	155712	44.5 ± 2.0	88.5 ± 1.4	39.3 ± 2.7	51.6 ± 2.5	98.4 ± 0.7
	TabDPT	43428	40.6 ± 1.7	85.7 ± 1.2	39.2 ± 3.1	47.4 ± 2.0	98.2 ± 0.7
	Aerial+	96837	40.5 ± 1.8	93.4 ± 1.7	19.6 ± 2.6	45.9 ± 2.1	98.3 ± 0.9
	RandomForest	112529	47.1 ± 2.1	94.7 ± 0.5	37.8 ± 1.8	53.9 ± 2.9	88.4 ± 3.5
	XGBoost	128256	29.3	77.7	25.0	36.4	95.2
	FP-G (0.5)	468403	62.4	91.1	29.9	68.8	92.8
	FP-G (0.3)	609836	47.1	90.9	28.3	53.8	98.9
	FP-G (0.2)	677858	38.6	91.3	28.8	45.3	99.8
	FP-G (0.1)	818765	29.8	91.7	27.1	35.5	100.0

C.7 Sample rules

Table 18 shows sample rules from the datasets used in the experiments. Each entry in the table shows both the rule in its raw form using column names in the datasets, a plain language translation, and core quality metrics. The plain language translations are derived using each datasets UCI documentation (Kelly et al., 2023). The rules are obtained using the seed 42 listed in Table 6.

Table 18: Sample rules covering all methods in the datasets used across all experiments. Each row illustrates the rules in their raw form, using the column names from the datasets, as well as plain language translations based on their documentation and rule quality scores.

Dataset	Method	Rule (raw / plain language)	Supp.	Conf.	Zhang
Acute Inflammations	TabPFNv3	`lumbar-pain=no AND micturition-pains=yes` → `bladder-inflammation=yes` No lumbar pain and painful urination → bladder inflammation	25.0	100	100
ASD Screen. (Adolescent)	TabICLv2	`jundice=no AND result=6` → `Class/ASD=NO` No history of neonatal jaundice and an AQ-10 score of 6 → negative ASD screening result	12.5	100	100
Breast Cancer	TabPFNv3	`inv-nodes=0-2 AND irradiat=no` → `node-caps=no` 0–2 positive axillary lymph nodes and no radiotherapy → no lymph-node capsule invasion	61.9	96.7	88.2
Cervical Cancer Behavior Risk	TabPFNv3	`intention_commitment=15 AND ca_cervix=0` → `behavior_sexualRisk=10` *intention_commitment* at 15 (its modal value) and no cervical cancer diagnosis → *behavior_sexualRisk* at 10 (its modal value)	45.8	100	100
Congressional Voting Records	Random Forest	`adoption-of-the-budget-resolution=y AND physician-fee-freeze=n` → `Class=democrat` Voted yes on the budget resolution and no on the physician fee freeze → Democrat	50.3	100	100
Fertility	TabPFNv3	`smoking=-1 AND season_discretized=(-1.001-0.33]` → `diagnosis=N` Never smoked, and semen analysis performed in winter or spring → normal semen diagnosis	34.0	97.1	78.4
Hepatitis	Random Forest	`Spiders=2 AND Ascites=2` → `Varices=2` Spider angiomata (a liver-disease skin sign) and ascites (abdominal fluid buildup) → esophageal varices	59.4	98.9	93.8
Mushroom	XGBoost	`gill-color=b AND spore-print-color=w` → `poisonous=p` Buff gill color and white spore print → poisonous	21.3	100	100
Spambase	TabDPT	`word_freq_650_discretized=(0.03-9.09] AND word_freq_85_discretized=(0.1-20.0]` → `Class=0` Contains the token “650” (personal, non-spam mail) together with the token “85” → not spam	7.5	99.7	99.6

C.8 Practical Guidance for Choosing a Framework Instantiation

The experimental settings in Section 4 and Appendix C have shown many tradeoffs of framework instantiations based on association strength, number of rules and execution time, data coverage, and downstream predictive task performance. The choice of instantiation depends on the practitioner’s priority. When the goal is to find strong associations in the data, RandomForest performs the best with a notable decrease in data coverage. RandomForest is followed by TabPFNv3, which does not have the data coverage issue. When downstream generalization matters most, i.e., using rules to make interpretable decisions, particularly on small datasets, TFMs are the strongest choice, as shown in Settings 1 and 2, and they require no per-dataset training, unlike other methods. When execution time or being able to run on large itemset universes are the priority, tree-based instantiations and Aerial+ are faster and do not require GPU resources (Setting 3). However, RandomForest trades off data coverage for association strength, and both RandomForest and XGBoost occasionally fail to produce enough rules for CBA even after lowering τ_a, whereas TFMs and Aerial+ reliably produce enough classification rules both for CBA and CORELS.

Robustness to experimental choices is another consideration when choosing an instantiation. The relative ordering of instantiations (Figure 2 and Table 3) is preserved at antecedent length $a = 3$ (Appendix C.6), and most instantiations are stable across different binning resolutions (3, 5 and 10), with TabDPT's rule count being the most sensitive to the choice of bin count (Appendix C.2). Across-seed variation is also higher on small datasets, and largest on datasets with very few rows such as Cervical Cancer Behavior Risk (Appendix C.1). TabPFNv3 and RandomForest have shown the smallest variance across different random seeds in rule quality, while Aerial+ is the most stable in downstream prediction tasks.

D Hyperparameter Analysis

This section analyzes the hyperparameters of Algorithm 2.

Setup. Each TFM is run on all 10 datasets with 10 random seeds, varying τ_a and τ_c thresholds. First, $\tau_c = 0.8$ is fixed and $\tau_a \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ is varied. Next, $\tau_a = 0.5$ is fixed and $\tau_c \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$ is varied. Algorithm 2 is executed per TFM-dataset pair up to line 12 to obtain $\hat{P}$, and the thresholds are applied to the same $\hat{P}$, ensuring comparable results across hyperparameter configurations.

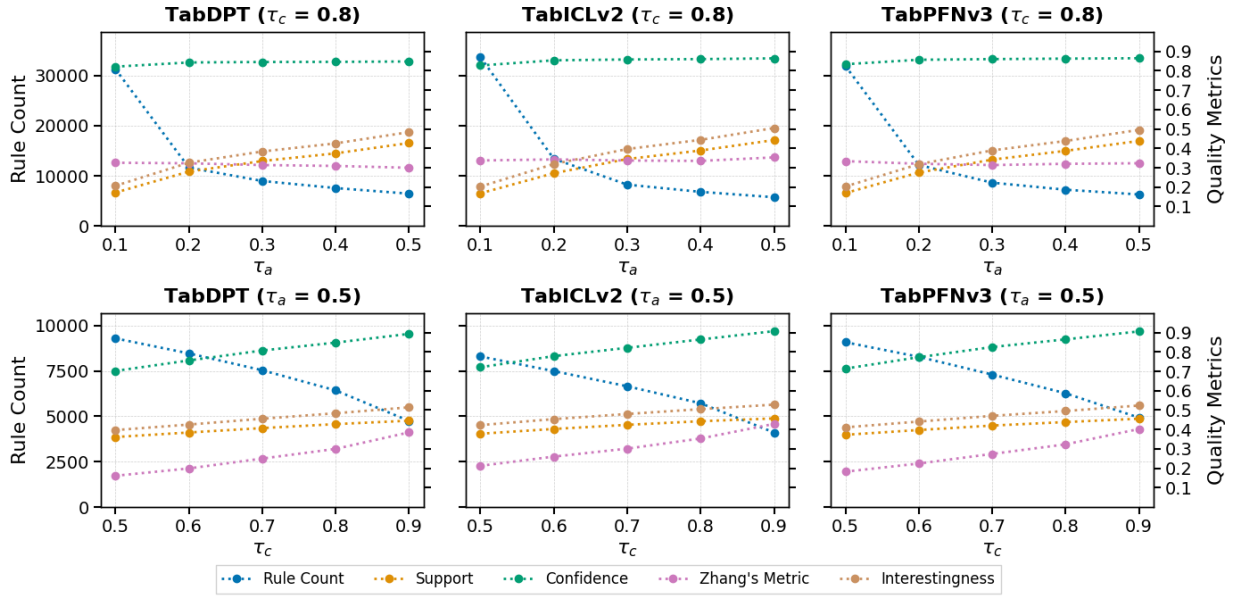


Figure 8: Hyperparameter analysis of Algorithm 2 over all 10 datasets. The left y-axis specifies rule counts, and the right y-axis specifies rule quality scores. Increasing τ_a (while keeping $\tau_c = 0.8$ fixed) leads to fewer rules with higher support and interestingness, while association strength (Zhang) and confidence do not change significantly. Increasing τ_c (while keeping $\tau_a = 0.5$ fixed) leads to higher support, confidence, association strength, and interestingness while reducing rule count.

The results of this analysis is presented in Figure 8. This figure shows the change in average rule counts and rule quality scores (support, confidence, Zhang's metric, and interestingness) as τ_a and τ_c change per TFM. The first row shows that as τ_a increases, rule count decreases while support and interestingness scores increase. No significant change is observed in confidence and association strength (Zhang) scores. The second row shows that as τ_c increases, support, confidence, association strength, and interestingness scores increase significantly, while rule count decreases.

The implications of the results are that τ_a impacts rule quality synonymously to the minimum support threshold of algorithmic methods, while τ_c has a more complex impact than the minimum confidence threshold. Hyperparameter settings depend on the rule learning task, as rules of different quality may lead to

different results across tasks. As an example, in an anomaly detection task, when considered as a classification task, it can be carried out with a rule-based classifier that should include rules of lower support. Therefore, lower τ_a values should be preferred. On the contrary, in a knowledge discovery task with the goal of finding interesting patterns in the data, both τ_a and τ_c should be set sufficiently high to optimize rule interestingness and association strength, while avoiding an excessively small number of rules.

E Frequent Itemset Mining with Tabular Foundation Models

As part of experimental setting 2 (Section 4.2), we utilized CORELS Angelino et al. (2018) to build rule-based classifiers for downstream task evaluation besides CBA for robust results. CORELS require frequent itemsets to build classifiers. Therefore, this section provides an adaptation of Algorithm 2 to learn frequent itemsets with TFMs.

The algorithm for learning frequent itemsets with single-target models (i.e., TFMs) out of the box is given in Algorithm 3. Differences to Algorithm 2 are marked with a green background color. Instead of two probabilistic antecedent and consequent thresholds, τ_a and τ_c, we now have a single similarity threshold τ_s. As in Algorithm 2, probe vectors for all itemset candidates of size up to a are generated once (line 2), and each feature's model is fit once and reused across all candidates (lines 3-8). The τ_s threshold is then checked against all features in a candidate itemset S_r (line 11), and itemsets whose marked features all exceed τ_s are saved and returned as the frequent itemsets (line 12).

Algorithm 3 Frequent itemset mining with single-target models

Require: Dataset D, single-target model $\mathcal{M}_\theta$, max antecedents a, similarity threshold τ_s
Ensure: Frequent itemsets $\mathcal{I}$

1: $\mathcal{I} \leftarrow \emptyset$, $F \leftarrow \text{GetFeatures}(D)$
2: $\mathbf{Q} \leftarrow \text{GenerateProbeVectors}(F, a)$, $\hat{\mathbf{P}} \leftarrow \mathbf{0}^{|\mathbf{Q}| \times m}$
3: **for all** feature $f_j \in F$ **do**
4: $\quad \mathbf{X}^{-j} \leftarrow \text{RemoveFeature}(D, f_j)$
5: $\quad \mathbf{y}^j \leftarrow \text{GetLabels}(D, f_j)$
6: $\quad \mathcal{M}_\theta.\text{fit}(\mathbf{X}^{-j}, \mathbf{y}^j)$
7: $\quad \mathbf{Q}^{-j} \leftarrow \text{RemoveFeature}(\mathbf{Q}, f_j)$
8: $\quad \hat{\mathbf{P}}[\cdot, f_j] \leftarrow \mathcal{M}_\theta.\text{predict_proba}(\mathbf{Q}^{-j})$
9: **for all** probe $\mathbf{q}_r \in \mathbf{Q}$ with itemset S_r **do**
10: $\quad$ **if** $\min_{i \in S_r} \hat{\mathbf{P}}[r, i] \geq \tau_s$ **then**
11: $\quad\quad \mathcal{I} \leftarrow \mathcal{I} \cup \{S_r\}$
12: **return** $\mathcal{I}$