Ensemble Techniques for PDFA Learning

Diversity-Driven Ensemble Learning with the Alergia Algorithm

Błażej Łytkowski

Responsible Professor: Sicco Verwer
Supervisor: Simon Dieck

EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 22, 2025

Name of the student: Błażej Łytkowski
Final project course: CSE3000 Research Project
Thesis committee: Sicco Verwer, Simon Dieck, Merve Gürel

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Abstract. Probabilistic deterministic Finite Automata (PDFA) learning is a machine learning method used for tasks requiring human understandability and more formal validation. In recent years we saw numerous applications of ensemble techniques with other machine learning models such as decision trees. Following the success of these attempts, in this paper, we aim to integrate ensemble methods into Alergia, which is a famous algorithm in the PDFA learning realm. We present a randomized variation of the Alergia algorithm and show how to build an ensemble out of it. Such an ensemble can visibly outperform a single Alergia model, which is documented by a series of experiments. Next, we present a custom distance metric measuring dissimilarity between a pair of Alergia models. We show how it can be used to build an Inter-Model Variety score quantifying the overall diversity of a group of models. Lastly, we analyze several methods that strive to select a well-performing diverse ensemble out of a big population of generated models.

1 Introduction

Probabilistic Deterministic Finite Automata (PDFA) learning is a branch of machine learning derived from Automata, which can be used for classification and prediction tasks. Its aim is to create a minimal PDFA consistent with the training data that can generalize to unseen test data. The inherent visualization potential of PDFAs makes them useful in areas requiring interpretable models such as software analysis [1] and anomaly detection [2].

Alergia [3] is a famous deterministic algorithm for learning a regular language given a data set consisting of only positive data - words that all belong to the language. At its core, the algorithm starts with a Prefix Tree Acceptor (PTA) of the samples and then repeatedly reduces it to smaller automatons that are similar within statistical uncertainty to the original PTA. The reductions are performed using heuristic merges, where if two states share enough similarity, they get combined.

Ensemble learning is a general term for methods that combine the predictions of several different models to make a joint decision. It follows from the concept of the wisdom of the crowd that tells us the average of many individual guesses should result in an accurate prediction [4]. Ensemble methods have been successfully used with various machine learning methods, including some that are closely related to PDFA learning like random forests [5]. Despite this, the use of ensemble techniques with PDFA learning has remained an unexplored theme.

In this paper, we show how the deterministic Alergia algorithm can be modified to form an ensemble of distinct models. We propose a method of introducing randomness into the algorithm and then argue for an approach of combining the predictions of the ensembled models. We combine these ideas in a parallelizable implementation of the ensemble using the FlexFringe project [2] as a basis.

We bring up the concept of ensemble diversity formalized by Wood et al. [6] as a hidden factor in the bias-variety tradeoff of a model. We investigate how model diversity, shown to elevate the predictive performance of ensembles [4], can influence the Alergia ensemble. We present the Inter-Model Variety (IMV) score as a metric to quantify the diversity of a group of Alergia-generated PDFAs. Furthermore, we show several methods to optimize for the ensemble diversity: a heuristic that maximizes the IMV score and clustering.

A particular focus of this research is testing ensemble performance under sparse training conditions, such as those encountered in real-life datasets like the HDFS software logs. In such scenarios, a single model trained with Alergia often struggles to generalize effectively.

We demonstrate that ensemble-based approaches offer a significant advantage in these settings.

The general structure of this paper is as follows. Section 2 gives an overview on the theoretical background of this research. In Section 3 we present how the original Alergia algorithm can be modified to produce an ensemble of models. Next, in Section 4, we experimentally analyze the performance of ensembles and optimizing for ensemble diversity. Section 6 summarizes the contributions of this paper.

2 Theoretical Background

In this chapter we introduce the theoretical background of the research. We begin in Section 2.1 by formally defining a PDFA. Next, in Section 2.2, we briefly go over the general concepts of evidence-driven merging and the Alergia algorithm. Lastly, in Section 2.3, we showcase Perplexity which is an evaluation metric used with probabilistic models.

2.1 Probabilistic DFA Extension

Deterministic finite automaton (DFA) is a machine with a finite number of states that for every string of symbols (trace) over some alphabet either accepts or rejects it. Probabilistic deterministic finite automaton (PDFA) is built on top of a DFA by assigning each transition of the machine some probability of occurring next, and every state a probability of being an accepting state. With such modifications, the machine can assign a probability to every trace over the alphabet, with some traces having a probability of zero.

Formally, we can define a PDFA as a DFA with an addition of a function $\pi : S \times \Sigma_+ \to [0, 1]$, which establishes the probabilities of transitions. It maps every combination of the current state $s \in S$ and a symbol $x \in \Sigma_+$ to a probability value, such that for every state s of the automaton, the sum of all values of this function is equal to 1. The extended alphabet Σ_+ is the regular alphabet plus the symbol ϵ marking the end of the trace. The value $\pi(s, \epsilon)$ indicates the probability of a trace finishing in state s.

The set of all traces over the regular alphabet is noted as Σ^*. Then, we define the probability of a trace $x \in \Sigma^*$ in a PDFA M as:

$$M(x = x_1 x_2 \dots x_n) = \Big(\prod_{i \in 1,\dots,n} \pi(s_{i-1}, x_i) \Big) \cdot \pi(s_n, \epsilon) \tag{1}$$

where s_i is the state reached after evaluating first i symbols of x. Because of the property that probabilities in every state sum up to 1, the following holds:

$$\sum_{x \in \Sigma^*} M(x) = 1 \tag{2}$$

Hence, a PDFA M can be also seen as a discrete probability distribution over the space Σ^*. In this context, performing random walks on M can be seen as sampling elements from the distribution.

2.2 PDFA Learning and Alergia

Evidence driven state merging [7] is one of the most prominent techniques in the domain of DFA/PDFA Learning. The algorithm starts with a Prefix Tree Acceptor of the input trace

data, which is simply a prefix tree build from the traces that also stores the frequency of traces going through and ending up in every tree node. The tree is reduced step-by-step to a minimal DFA by applying consistent merges. The key feature of the method is that it orders all the possible consistent merges by the amount of evidence in the data that supports them. The optimal version of the algorithm always chooses the best scored merge.

Alergia [3] is a probabilistic variant of the evidence-driven state merging framework, where the criterion in which the merge quality is ordered is a probabilistic bound. In short terms, the algorithm considers all merges which lead to models that could have produced the training data within some probabilistic uncertainty. The regular version of the algorithm always select the most probable of all these merges, which makes it deterministic. The algorithm finishes when there are no more merges that fulfill the probabilistic bound.

2.3 Evaluating Probabilistic Models

How can the quality of probabilistic models such as a PDFA be evaluated and compared? In case of PDFA models, this question primarily comes down to the predictive performance of a model on unseen test data. The computational footprint and memory requirements needed to train and deploy a model are usually minimal, because in general PDFA learning is regarded as a lightweight machine learning approach [8]. Furthermore, training and running an ensemble of independent models can be easily parallelized and scales linearly with ensemble size.

When it comes to evaluating the predictive performance of the models, the problem reduces to comparing probability predictions for a set of traces to ground truth values. To formalize, we seek a metric that given a set of target probabilities P and a set of predicted probabilities Q, outputs a numeric score indicating the quality of the predictions in comparison to the target. To handle this, Verwer et al. [9] employed the Perplexity score, a standard metric in probabilistic modeling that quantifies how well a probability distribution predicts a sample. Intuitively, it measures the average "surprise" the model experiences when seeing the actual data. The larger the perplexity, the less likely the model is to guess a sample from the real distribution. The metric is defined as:

$$\text{Perplexity(p, q)} = 2^{\sum_x p(x) \cdot \log q(x)} \tag{3}$$

where p is the target distribution and q is the proposed distribution. The important fact here is that p and q must have the same domain and that the metric sums over all the elements in the domain. To make the metric applicable for PDFAs, that often give a non zero probability value to an infinite set of traces, we compute the perplexity over a finite sample set of test traces. Given an alphabet of symbols Σ and a test set of unique traces over the alphabet $TS \subset \Sigma^*$, we define it as:

$$\text{Score(p, q, TS)} = 2^{\sum_{x \in TS} p(x)/N_p \cdot \log q(x)/N_q} \tag{4}$$

In this formula we added terms N_p and N_q as normalization constants equal to the sum of $p(x)$ and $q(x)$ on all the traces in the test set. This is done to normalize the value of the metric between different test sets.

If $q(x)$ was misaligned with the domain and predicted $q(x) = 0$ for some positive trace in the test set, perplexity would be infinitely high. We can punish the models that don't recognize all the test traces, while not setting their score to infinity by replacing the zero predictions with the average of all other guesses on the test set. This approach is fair as

only information available to the predictor is used, and the assumption that each test trace is positive is known by all.

3 Combining Alergia and Ensembles

This chapter aims to explain the analysis of the main research problem and guides the reader through the implementation of the solutions. It starts in Section 3.1 by showing how the randomization of Alergia algorithm was achieved and then in Section 3.2 presents how the predictions of an ensemble of models are combined. Next, the Section 3.3 introduces the Inter-Model Variety (IMV) score used to quantify the predictive quality of one model in relation to another one. Section 3.4 continues by showing how a true distance metric can be achieved using the IMV score. The chapter ends with Section 3.5 presenting how the IMV score and a chosen clustering technique can be used to prune a huge collection of models into a similarly performing smaller ensemble.

3.1 Randomization of Alergia

The key research question of this paper is how to modify or build upon the original Alergia algorithm to produce a varied ensemble of models. Among the many general methods for transforming a single model into an ensemble, one of the most universal and easy to implement is the introduction of randomness into the learning process. This research adopts that method to achieve model diversity without straying from the core principles of Alergia. The added benefit of this approach is the direct integration with other known variations or improvements of the Alergia algorithm. Existing problem-solving strategies built around Alergia can thus be easily converted to work with our ensemble framework.

To introduce randomness into the Alergia algorithm, we modify the merge selection process, which allows for the creation of a diverse group of models using the same training data. Specifically, we individually skew the quality score of each candidate merge by multiplying it with a number $\boldsymbol{x \leftarrow U(1-r, 1)}$. Here, $\mathbf{U}$ is a uniform distribution from $\mathbf{1-r}$ to $\mathbf{1}$.

This method allows control over the amount of randomness introduced into the learning process. For example, one can increase the randomization when training on smaller datasets, where the number of probable models is limited. Larger values of the parameter r lead to greater randomness in the merge decisions, and bigger search space of possible models.

3.2 Ensemble Voting

A central question when designing an ensemble is how to combine the outputs of its component models. In our case, each randomized Alergia model produces a probability between 0 and 1, allowing for a natural aggregation method: taking an average of the predictions. The simplest version of this is uniform averaging, where all models have equal weights.

We adopt this equal-weight strategy deliberately. In sparse training sets scenario, one of the focuses of this work, we want to amplify the benefit of model diversity. Validation data, sourced from sparse training set, cannot reliably distinguish truly generalizable solutions from accidental ones. Thus, adjusting weights based on signals coming from the training data can reduce the diversity among models, effectively pulling the ensemble closer to a single overfitted solution.

By contrast, equal weighting respects the independence of the diverse hypotheses generated by randomized training. It treats each model as a valid interpretation of the data,

which is especially important in high-uncertainty, low-data settings. Prior work in ensemble learning also supports this view: uniform averaging often performs surprisingly well when model errors are uncorrelated or weakly correlated [10]. In our setting, the ensemble's strength comes from the varied inductive biases of its members, not from overfitting a small validation set.

3.3 Inter-Model Variety Score

Another area explored in this research is a well-known phenomenon in ensemble learning, that increased diversity among ensemble members can lead to improved performance. This has been theoretically grounded by Wood et al. [6], who identify diversity as a hidden dimension of the bias-variance tradeoff and show that greater diversity directly contributes to a reduction in predictive loss. We propose a method of quantifying ensemble diversity called the Inter-Model Variety (IMV) score. The IMV score enables comparison of structural differences between ensembles and provides an optimization objective for ensembling Alergia models.

The construction of the IMV score is based on measuring for each pair of models in the ensemble how differently they behave. One reasonable way to do this is to evaluate how well one model predicts the outputs of another. This leads to a scoring method based on **sample cross entropy**, which we define as a directional measure of how much model A disagrees with model B on a sample of traces. The idea leverages the fact that a PDFA model can both generate and evaluate the probabilities of traces. In practice, we sample a set of traces from one model, and then use the other model to evaluate them. A higher value of sample cross entropy indicates that the two models make different predictions, thus capturing model dissimilarity.

This approach is closely related to the perplexity metric discussed earlier in Section 2.3, which we used to evaluate a model's performance on test data. In both cases, we compare a model's probability estimates to a set of target values. However, in sample cross entropy, we omit the exponentiation step used in perplexity. This allows us to measure disparity in a more linear scale. It is important to note that sample cross entropy is not symmetric, and that a model compared with itself does not necessarily yield zero.

We define the sample cross entropy from model A to model B as follows:

$$h(A \to B) = \sum_{x \in S_k(B)} \frac{B(x)}{N_B} \cdot \log \frac{A(x)}{N_A} \tag{5}$$

Here, $S_k(B)$ denotes a sample of k unique traces generated by model B, with $B(x)$ and $A(x)$ referring to the probabilities assigned to trace x. N_A and N_B are normalization constants. The resulting value, $h(A \to B)$, gives us a directional score reflecting how surprising the behavior of model B is to model A, over a sample of B's trace distribution. This allows us to construct a set of pairwise distances between models and ultimately define the Inter-Model Variety of an ensemble.

To compute the Inter-Model Variety (IMV) score for an ensemble E of n models, we begin by generating a unique sample set $S_k(M)$ of traces for each model $M \in E$. Then, for each ordered pair of models $(A, B) \in E \times E$, we compute the sample cross entropy $h(A \to B)$ using the traces from $S_k(B)$. We normalize these values by subtracting the self-cross entropy of the reference model. This gives us the normalized score:

$$f(A, B) = h(A \to B) - h(B \to B) \tag{6}$$

The IMV score is then defined as the sum of all these normalized distances:

$$\text{IMV(E)} = \sum_{A,B \in E \times E} f(A, B) \tag{7}$$

This value aggregates all pairwise comparisons, treating each model both as an evaluator and as a source of behavior. The score grows with the diversity of model predictions across the ensemble, rewarding ensembles whose members strongly disagree on each other's sample outputs.

The intuition behind this approach is that if all models perform similarly and make similar predictions, the pairwise distances will be low, resulting in a low IMV score. In contrast, if models behave very differently from each other, then the distances will be high, and the IMV will reflect this. While the score doesn't take into account different performance levels across models and cannot be used to compare ensembles of different sizes, it still offers a simple and effective way to quantify model diversity, which can be useful for understanding and optimizing ensemble design.

3.4 Pairwise Model Distance Metric

Having defined the notion of sample cross entropy of one model in relation to another model (Equation 5), we are able to build a true distance metric out of it. A distance metric is useful in understanding the search space of possible models given some training data and enables us to visualize an ensemble with its various components in this space. We define a distance metric for a set of n models M as a function $d : M \times M \to \mathbb{R}$ with the following properties:

1. Symmetry: for all $i, j \in M \times M$ it holds that $d(i, j) = d(j, i)$.
2. Non-negativity: for all $i, j \in M \times M$ it holds that $d(i, j) \geq 0$.
3. Zero self-distance: for all $i \in M$ it holds that $d(i, i) = 0$.

One candidate that can be used as a base for a metric fulfilling the above requirements is a simple measure of similarity proposed in [11]. The similarity measure is based on cross entropy for discrete probability mass functions and is defined as:

$$Sim(A, B) = \frac{h(A \to A) + h(B \to B)}{h(A \to B) + h(B \to A)} \tag{8}$$

with $h(A \to B)$ being the sample cross entropy from Equation 5. The similarity measure is (1) symmetric, (2) greater than zero, and (3) achieves the maximum value in a self comparison $Sim(A, A) = 1$. We define a metric having the distance metric properties as:

$$Dis(A, B) = \frac{1}{Sim(A, B)} - 1 \tag{9}$$

This metric allows us to visually represent the different relations between models of an ensemble with techniques such as Multidimensional Scaling, which can reduce a high dimensional matrix of distances between points to a 2D plot. This can be seen in Figure 3 of Appendix A. Furthermore, a distance metric also enables methods such as clustering of ensemble models, which we elaborate on in Section 3.5.

3.5 Ensemble Pruning

The main advantage of the algorithm described in Section 3.1, which can independently generate multiple ensemble models, is its inherent parallelism. However, simply averaging over a large number of randomly generated models eventually reduces the diversity of the ensemble: the randomness becomes diluted, and the ensemble starts to converge towards a central tendency. This effect can reduce the overall generalization ability and lead to overfitting on the test data. Another motivation for limiting the ensemble size is to preserve one of the core advantages of PDFA learning : its lightweight nature compared to more computationally demanding methods like neural networks.

To keep the benefits of exploring a big region of the search space by generating a lot of models, while staying with compact ensemble size, we employ **ensemble pruning**: selecting a subset of all the generated models. For this to be effective, the pruning method must be capable of selecting a subset that performs better than a randomly generated small ensemble. In this work we focus on diversity-driven pruning, where the objective is to incentivize differences between the hypotheses within the final ensemble. A diverse ensemble helps correct for the individual errors of its models, as supported by ensemble learning theory [10].

We consider three pruning methods, each designed to either maximize the diversity of the ensemble directly (using IMV) or ensure representative coverage of the model space (using clustering). These two perspectives focus on different but related goals: maximizing IMV works towards selecting the most distinct models, while clustering strives to accurately showcase the whole search space of models by sampling from its natural groupings.

IMV-based Greedy Heuristic The first pruning method we use is a heuristic aimed at maximizing the IVM score of the pruned ensemble. By choosing the most diverse models as our ensemble, we assume that it will represent all the extreme hypotheses, thus spanning the whole search space. Furthermore, we hope that the average of all the extreme models will accurately predict the middleground of the search space, so traces that get similar predictions from most of the models.

Formally, we want to choose a subset of size n of the initial group: $S \subset E$, such that the IVM value of the smaller ensemble: $\sum_{(A,B)\in S\times S} f(A,B)$ is maximized. This is a known NP-Hard problem, but we can use a simple heuristic that step by step removes a single model C from the big ensemble, for which the sum $\sum_{A\in S} f(A,C) + f(C,A)$ is the smallest, until we reach the desired ensemble size. This is illustrated in Algorithm 1.

Algorithm 1 Ensemble pruning with IVM

Require: $m \leftarrow$ initial ensemble size
$n \leftarrow$ desired ensemble size
$H \leftarrow$ normalized pairwise cross entropies as a matrix
Ensure: $H.size = m \times m$
  $selection \leftarrow list[1, 2, \ldots, m]$
  while $length(selection) > n$ **do**
    $scores \leftarrow H.sumRows() + H.sumColumns()$
    $i \leftarrow minIndex(scores)$
    $selection \leftarrow selection.removeIndex(i)$
    $H \leftarrow H.removeRow(i)$
    $H \leftarrow H.removeColumn(i)$
  return $selection$

Clustering The second pruning approach is based on clustering the full ensemble to identify representative models for the pruned ensemble. The goal is to avoid overfitting to the implicit probability distribution of models generated from a particular training set.

As the ensemble size grows, the distribution of its models converges to an implicit distribution centered on the most likely interpretations of the data. This means that high-probability models, closely aligned with the training set, will be much better represented in the ensemble than alternative hypotheses. To increase diversity, we cluster the ensemble and select one model from each cluster, giving equal representation to each of the present hypotheses.

We use two clustering methods: **K-Medoids** and **Affinity Propagation**. K-Medoids is an alternative to K-Means which, instead of computing centroids, selects k actual models as cluster centers. Additionally, K-Medoids does not require a Euclidean distance, so we can directly use the custom distance metric defined in Section 3.4. K-Medoids is especially suitable for spatial-style clustering, which assumes that clusters are compact and roughly Gaussian-shaped in the distance space. This makes it well-suited to datasets where models form dense, clearly separated groups. However, in cases where model similarity relationships form less regular or overlapping structures, K-Medoids may not accurately reflect structure of the ensemble.

Affinity Propagation, on the other hand, is a message-passing algorithm that selects clusters based on pairwise similarities without requiring the number of clusters to be pre-specified [12]. It builds a network of mutual affinities between models and identifies central exemplars that best summarize the rest of the data. Unlike K-Medoids, Affinity Propagation is not restricted to finding spatially compact clusters, and can capture irregularly shaped or even nested cluster structures. This makes it particularly effective when the diversity in the ensemble is more structural or topological than spatial, as is often the case with Alergia models and our custom distance metric. We include Affinity Propagation as a complementary approach to K-Medoids, as its emphasis on network structure over distance shape might yield more representative and diverse ensembles in this setting.

4 Experimental Setup and Results

This chapter presents the findings and insights gained from the experimental phase of this research. Section 4.1 details the experimental setup, selection of datasets and used evaluation methods. Section 4.2 compares the performance of a single Alergia model against basic ensembles. Next, section 4.3 explores the effect of advanced ensemble techniques - pruning. Finally, Section 4.4 presents the application of the Alergia ensemble to a real-world problem, using software logs for anomaly detection.

4.1 Data Selection and Evaluation

To thoroughly evaluate the proposed ensemble methods, experiments were conducted on three types of data: (1) the Reber grammar [13], (2) randomly generated PDFAs using the PAutomaC methodology [9], and (3) real-world software traces from the HDFS dataset [14, 15]. This combination provides a complete testing environment. Artificial data allows for controlled and adjustable experiments with direct comparison against a known ground truth. Real-life data serves as a practical benchmark for model performance in realistic conditions with unknown underlying model.

Artificial Data: Random PDFAs The artificial datasets were generated by first constructing a random target PDFA using the PAutomaC methodology. The training set consisted of traces generated by random walks over the automaton. The test set was composed of unique traces also generated via random walks. If a generated trace was already present in the test set, a new walk was performed to maintain uniqueness.

Each test trace was annotated with its true probability, calculated as the product of the transition probabilities encountered during the walk. This ground truth allowed a precise evaluation of the learned models by computing the **perplexity score**. The test set size was chosen to be comparable to the amount of data typically needed for a single model to reliably approximate the target.

Real-World Data: HDFS Software Logs The second dataset consisted of real-world HDFS system logs. This dataset is a standard benchmark in anomaly detection research. Each trace in the dataset represents a sequence of event types, labeled either as `Normal` or `Anomaly`. Only `Normal` traces were used to train the PDFA models, which reflects the unsupervised nature of most real-world training conditions.

To evaluate model performance, the remaining `Normal` traces were mixed with anomalous ones to form a test set. Each trace was assigned a probability by the learned model, where traces with probability under some threshold are regarded as anomalies. The effectiveness of the ensemble in distinguishing between the two classes was assessed using the Precision-Recall Curve (PRC) based on the value of the detection threshold.

4.2 Comparison of Single Model and Ensemble Performance

This section compares the predictive performance of a single deterministic Alergia model with that of a randomized ensemble of Alergia models. The comparison focuses on how these models behave under varying training set sizes: levels of sparsity. For ensembles, the perplexity results are averaged over 50 independent training runs, with standard error bars shown to illustrate variance.

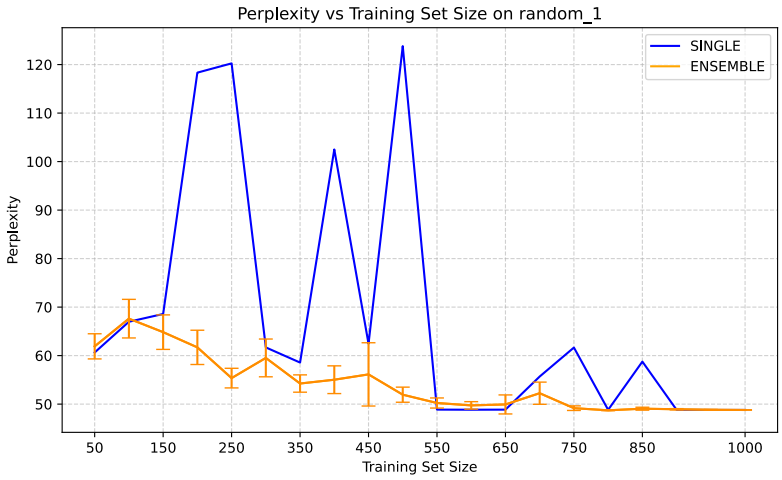

Fig. 1. Single model vs an ensemble performance against growing training set size. The perplexity score and standard deviation comes from 50 independent train runs on training set of each size.

Performance Trends Across Training Set Sizes Figure 1 presents the performance of both the single model and the ensemble as the size of the training set increases. Each training set is constructed by appending new traces to the previous set, ranging from very sparse (50 traces) to dense (1000 traces). The test set remains fixed across all experiments.

The ensemble shows a clear and steady convergence trend as more training data is added. Even at low sparsity levels, the ensemble behaves steadily, with lower perplexity and less variance compared to the single model. The performance of the single model changes in sharp jumps, suggesting that it either fails or succeeds in capturing the underlying structure of the data depending on the composition of the training set.

Aggregate Comparison Across Datasets To generalize the observations beyond a single learning curve, we evaluate model performance on the synthetic datasets under two training settings: sparse and dense training set. Table 1 summarizes the results. The ensemble outperforms the single model in all sparse settings, often by a significant margin. In dense settings, where training data is plentiful, the gap between the two methods narrows.

These results support the hypothesis that ensembles are particularly worthwhile when training data is sparse relative to model complexity. We hypothesize that in such cases, the single model tends to overfit, converging to narrow, suboptimal interpretations of the data. The ensemble mitigates this by combining diverse hypotheses, leading to a more robust and broad approximation of the target model.

This behavior aligns with previous findings in ensemble learning literature. Ensembles often improve generalization by reducing model variance and averaging over multiple decision boundaries [4]. Our results demonstrate that these benefits also apply in the context of PDFA learning using Alergia, particularly in data-scarce scenarios.

A surprising phenomenon occurs in the case of the `random_4` dataset. Here, the models in the dense training set scenario actually perform worse than in the sparse scenario. Our presumption is that the dense training set size is actually too small to generate small, high-confidence models, and instead produces large models with few merged nodes and high variance.

Dataset	**Train size**	**Best score**	**Avg. perplexity increment to best**		
			Single	Ensemble ($r = 1.5$)	Ensemble ($r = 3.0$)
reber	35	50.6	113.7 ± 250.1	29.6 ± 40.2	19.7 ± 27.5
	80	50.5	1.2 ± 0.8	1.4 ± 1.0	2.2 ± 1.6
random 1	300	49.3	43.9 ± 53.1	8.5 ± 6.4	8.7 ± 5.3
	900	48.6	22.5 ± 51.9	0.7 ± 1.7	0.6 ± 0.8
random 2	250	81.3	99.4 ± 67.7	24.7 ± 15.1	26.3 ± 10.3
	700	68.8	7.8 ± 13.4	1.6 ± 2.4	1.9 ± 1.1
random 3	350	119.1	410.6 ± 223.7	113.7 ± 80.8	130.8 ± 59.4
	1000	98.7	51.8 ± 37.6	12.8 ± 9.2	9.1 ± 5.8
random 4	150	20.6	26.7 ± 32.3	9.0 ± 7.5	8.0 ± 7.7
	600	22.4	40.4 ± 35.2	18.9 ± 9.3	16.4 ± 9.3

Table 1. Single model vs ensemble performance across datasets. Values represent perplexity difference to the best score on the given trainset. The scores are averaged over 50 independent runs with different training. **r** is the value of the random parameter. Both ensemble models have a size of 20.

4.3 Effects of Optimizing for IVM with Ensemble Pruning

To understand the effect various ensemble pruning techniques presented in Section 3.5 have on the performance of the ensemble, we conducted a series of experiments. These experiments aim to answer a central research question: how can ensemble diversity, be harnessed to improve the predictive performance of PDFA models trained with the Alergia algorithm? We evaluate this question using our IMV score as a way to quantify ensemble diversity, and test the impact of optimizing for diversity through pruning.

We consider the three pruning strategies described in Section 3.5: Max-IMV, K-Medoids clustering and Affinity Propagation. They are compared against the full ensemble they were created from and and a random ensemble of the same size as the pruned ones.

Trends in the Behavior of Pruned Ensembles In the first experiment, we analyze how the pruning method behave as the size of the generated models increases. Figure 2 shows representative runs highlighting trends and edge cases.

In the plot for `random_3`, we observe that both Max-IMV and clustering require a warmup period in the number of generated models to stabilize performance. This behavior is expected as with too few models, pruning has limited selection power and high variance in outcome. Once enough models are available, some of the pruned ensembles begin to match or even slightly exceed the performance of the full ensemble.

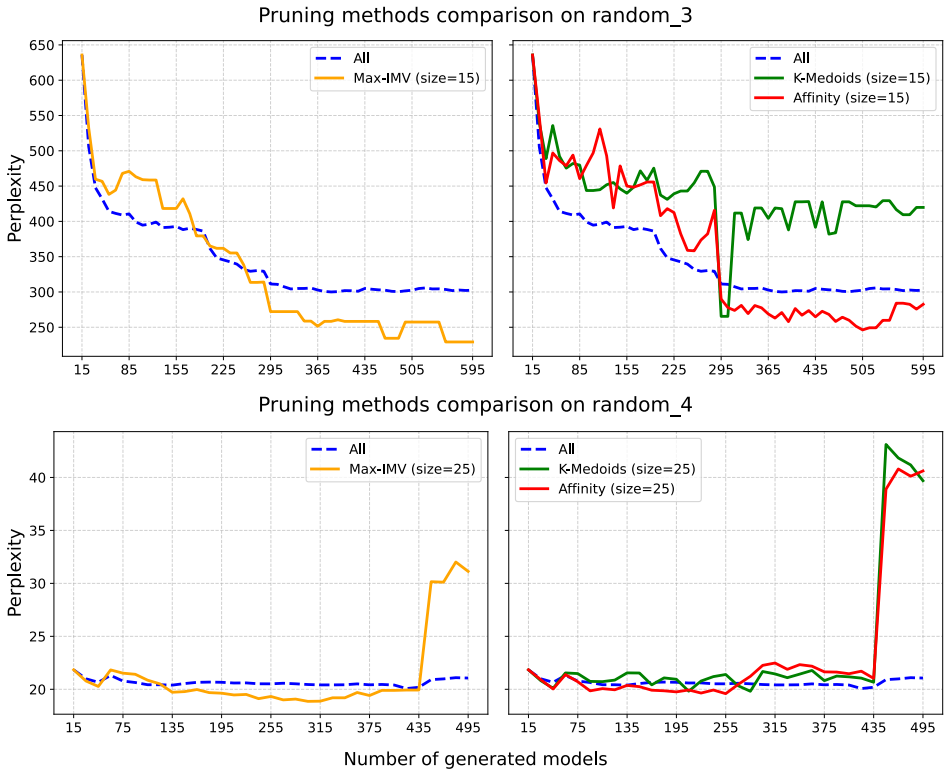

Fig. 2. Performance of various pruning methods versus an ensemble of all models on two datasets with a growing number of generated models.

An interesting behavior appears when the large ensemble is unbalanced due to one or more outlier models with extreme predictions, which is showcased in the plot for the dataset `random_4`. In such cases, clustering-based methods and Max-IMV often select these outlier models due to their uniqueness, which leads to a sharp spike in perplexity. In contrast, the full ensemble is more robust in these situations, as it dilutes the effect of any single model by averaging over the entire distribution. This suggests that while diversity can boost generalization, it must be balanced in terms of bias and variance.

Statistical Comparison Across Datasets To generalize the findings, we conducted a second experiment across three synthetic datasets. For each dataset, we trained a large ensemble of 200 models and pruned it down to 20 models using each of the four pruning methods. Each experiment was repeated 10 times on different training sets, and results were averaged. The results are summarized in Table 2.

Dataset	**Train size**	**Best score**	**Average perplexity difference to best**				
			All models	Random	Max-IMV	K-Medoids	Affinity
random 1	300	61.5	9.7 ± 5.1	14.9 ± 10.1	13.2 ± 8.8	12.5 ± 9.6	12.8 ± 10.2
random 2	300	98.1	12.2 ± 6.8	18.3 ± 10.2	8.7 ± 4.7	10.6 ± 6.7	9.3 ± 6.6
random 3	400	151.7	95.1 ± 49.8	113.9 ± 73.1	57.8 ± 39.8	97.1 ± 65.8	86.8 ± 61.5

Table 2. 200-model ensemble vs 20-model pruned ensembles performance across datasets. The scores are mean differences to the best score for a given dataset, accompanied by the standard deviation. The best score for each training set is highlighted.

Several key observations can be made:

- All pruning methods consistently outperform random selection. This confirms the role diversity plays in creating more informative ensembles than arbitrary selection.
- On more difficult datasets (`random_2` and `random_3`), pruned ensembles often outperform the full 200-model ensemble. This suggests that selecting a diverse, focused subset can improve generalization by filtering out redundant or overrepresented models.
- Max-IMV performs best on the more difficult datasets. We hypothesize this is because those datasets provide limited coverage of the full model space, and maximizing IMV encourages exploration of different regions of that space.
- Clustering-based methods perform best on the easier dataset (`random_1`), where the full ensemble already spans much of the model space. In this case, clustering can effectively find a representative subset of models.

These results support the hypothesis that ensemble diversity can meaningfully improve predictive performance in sparse-data PDFA learning. They also suggest an interesting tradeoff: while maximizing diversity is useful in sparsely explored search spaces of models, clustering is more effective when diversity is already present. These findings align with the diversity-bias-variance interpretation proposed by Wood et al. [6], where diversity complements bias reduction and variance control in ensemble generalization.

4.4 Real-World Anomaly Detection with the Alergia Ensemble

We evaluated the performance of ensemble and pruning on the HDFS software trace dataset. We tested a single model, 20-model ensemble, and another 20-model ensemble pruned down from 200 models. Models were trained on increasing volumes of normal traces and evaluated on a test set mixed with anomalous traces. Recall was measured at three high-precision levels. We showcase the results in Table 3.

Train size	**Model**	**Recall values for different precision levels**		
		$p = 0.99$	$p = 0.999$	$p = 0.9999$
200	Single	0.931	0.931	0.931
	Ensemble	0.974	0.579	0.579
	Max-IMV	0.984	0.984	0.984
500	Single	0.963	0.963	0.963
	Ensemble	0.986	0.709	0.709
	Max-IMV	0.986	0.986	0.986
1,000	Single	0.972	0.972	0.968
	Ensemble	0.991	0.991	0.984
	Max-IMV	0.992	0.992	0.984
5,000	Single	0.984	0.983	0.976
	Ensemble	0.998	0.995	0.992
	Max-IMV	0.998	0.995	0.992
100,000	Single	0.987	0.959	0.946
	Ensemble	0.995	0.972	0.966

Table 3. Recall (fraction of detected anomalies) at fixed precision levels **p** (fraction of flagged traces that are truly anomalous) for different model types and training set sizes on the HDFS dataset. The best and second-best score for each precision level is highlighted.

The results show a consistent trend: both regular ensembles (for larger training sizes) and Max-IMV pruned ensembles outperform the single model. In particular, the pruned ensembles achieve the best recall values across all precision levels—surpassing even the ensemble trained on 100,000 traces.

A key observation is how recall degrades with increasing precision thresholds. For small training sizes (e.g., 200 and 500), regular ensembles experience a sharp drop in recall as the precision level increases. By contrast, pruned ensembles maintain high recall even under the strictest precision requirements.

This suggests an important difference in ensemble behavior. Models whose recall collapses at high precision levels likely exhibit greater variance or internal inconsistency. In random ensembles, this can occur when overrepresented or misaligned models distort the output probabilities. The pruned ensemble, however, is explicitly selected for structural diversity, which helps avoid such skewed groupings. Diverse models may cover more of the search space, improving the model's balance and ability to distinguish subtle anomalies under high confidence thresholds.

These results indicate that using ensembles of models, especially with pruning, can improve anomaly detection even in sparse-data scenarios.

5 Responsible Research

This chapter outlines the ethical consideration associated with PDFA learning. Additionally, it mentions how the results of the experiments can be reproduced.

5.1 Ethics of Ensemble Alergia Learning

The introduction of ensemble methods into PDFA learning with focus on the Alergia algorithm, brings about several ethical considerations worth analyzing.

Environmental Impact While ensemble learning typically involves training and evaluating multiple models—potentially dozens instead of a single automaton—the overall computational footprint remains small. This is especially true when compared to large-scale neural network training, which often requires vast GPU resources and significant energy consumption. PDFAs are symbolic, compact, and interpretable models that can be trained efficiently on CPUs, making ensemble-based approaches in this domain highly energy-efficient. Thus, the environmental impact the presented approach remains minimal and aligns with goals of sustainable and responsible machine learning.

Misuse Potential As with many data-driven models, there exists a potential for misuse. If ensembles of PDFAs are applied to sensitive domains, such as modeling user behavior or predicting actions, they could inadvertently be used for surveillance, profiling, or manipulative user modeling without consent. While the models themselves are interpretable and relatively simple, the ethical responsibility lies in how they are applied and the nature of the data used for training.

Bias and Transparency Ensembles may also obscure interpretability to some extent, especially if conclusions are drawn from aggregate predictions without understanding individual automaton behavior. While PDFAs are inherently more transparent than black-box models, using an ensemble may require additional mechanisms to ensure that decisions made by the system remain explainable and auditable. Furthermore, any biases present in the training data may be consistently reinforced across ensemble members, which underlines the importance of ethical data sourcing and evaluation.

5.2 Reproducibility of Experiments

Parts of the data used in the research come from publicly available sources. The one exception is self generated random PDFAs, which are attached as model descriptions in `.dot` files. These can be parsed and rendered as diagrams by most software dealing with graph data. The parsed machines can be used to reproduce data sets similar to the ones used for the experiments. The code used in the research is also published alongside the paper. All of the experiments were performed on a 12 core personal computer and took a few hours at longest, thus reproducing them should be available to practically any willing researcher.

6 Conclusions and Future Work

This work explored how ensemble learning techniques can be applied to Probabilistic Deterministic Finite Automata (PDFA) learning using the Alergia algorithm.

We proposed a randomized variation of the Alergia algorithm by introducing controlled randomness into the merge selection process. This allowed the generation of a diverse set of models from the same training data. The resulting models were combined using uniform voting, which produced stable probabilistic predictions. Because the models are trained independently, both model generation and prediction can be parallelized efficiently. This approach proved effective and practical, forming the foundation of the ensemble framework studied in this work.

We introduced a distance metric based on cross entropy to compare structural differences between PDFA models. Building on this, we defined the Inter-Model Variety (IMV) score to quantify the diversity of a group of models. Our experiments showed that encouraging diversity, using either IMV maximizing heuristics or clustering, can help select a compact and diverse subset of models from a larger ensemble. In particular, the Max-IMV pruning method frequently outperformed both the full ensemble and random subsets, especially in low-data scenarios. However, we also observed that if the ensemble contains an extreme or misaligned model, diversity-maximizing strategies may amplify its influence. This suggests a need for future work on stabilizing diversity-based pruning methods, possibly by incorporating model quality evaluation or outlier detection.

Our experiments demonstrated that ensembles consistently outperform single Alergia models when training data is sparse. Moreover, ensemble pruning can further improve predictive performance while reducing ensemble size. In the HDFS anomaly detection experiments, pruned ensembles even outperformed full ensembles trained on 100 times more data, indicating that smart model selection can substitute for large data volumes.

This research opens several directions for further exploration. First, the impact of hyperparameters such as the level of randomization and ensemble size could be studied in more detail. Second, the IMV score could be combined with other metrics to balance diversity and accuracy during pruning.

In conclusion, ensemble methods, when carefully constructed and optimized for diversity, offer a significant performance boost to PDFA learning with Alergia. They improve generalization, especially in sparse-data settings, and fill a gap in lightweight high-performing machine learning methods for anomaly prediction.

References

[1] Paul Fiterau-Brostean et al. "Model Learning and Model Checking of SSH Implementations". In: *SPIN'17: PROCEEDINGS OF THE 24TH ACM SIGSOFT INTERNATIONAL SPIN SYMPOSIUM ON MODEL CHECKING OF SOFTWARE*. 24th ACM SIGSOFT International SPIN Symposium on Model Checking of Software (SPIN). Ed. by H. Erdogmus et al. New York: Assoc Computing Machinery, 2017, pp. 142–151.

[2] Sicco Verwer et al. *FlexFringe: Modeling Software Behavior by Learning Probabilistic Automata*. Mar. 28, 2022. URL: `https://arxiv.org/abs/2203.16331v4`.

[3] Rafael C. Carrasco et al. "Learning Stochastic Regular Grammars by Means of a State Merging Method". In: *Grammatical Inference and Applications*. Ed. by Rafael C. Carrasco et al. Berlin, Heidelberg: Springer Berlin Heidelberg, Nov. 23, 2002, pp. 139–152.

[4] Omer Sagi et al. "Ensemble learning: A survey". In: *WILEY INTERDISCIPLINARY REVIEWS-DATA MINING AND KNOWLEDGE DISCOVERY* 8.4 (Aug. 2018), e1249.

[5] Simon Bernard et al. "Dynamic Random Forests". In: *PATTERN RECOGNITION LETTERS* 33.12 (Sept. 1, 2012), pp. 1580–1586.

[6] Danny Wood et al. "A Unified Theory of Diversity in Ensemble Learning". In: *Journal of Machine Learning Research* 24.359 (2023), pp. 1–49.

[7] Wojciech Wieczorek. "State Merging Algorithms". In: *Grammatical Inference: Algorithms, Routines and Applications*. Cham: Springer International Publishing, 2017, pp. 19–31.

[8] Robert Baumgartner et al. *PDFA Distillation via String Probability Queries*. June 28, 2024. arXiv: `2406.18328[cs]`.

[9] Sicco Verwer et al. "Results of the PAutomaC Probabilistic Automaton Learning Competition". In: *Proceedings of the Eleventh International Conference on Grammatical Inference*. International Conference on Grammatical Inference. ISSN: 1938-7228. PMLR, Aug. 16, 2012, pp. 243–248.

[10] Thomas G. Dietterich. "Ensemble Methods in Machine Learning". In: *Multiple Classifier Systems*. International Workshop on Multiple Classifier Systems. Springer, Berlin, Heidelberg, 2000, pp. 1–15.

[11] Xiaozhuan Gao et al. "Cross entropy of mass function and its application in similarity measure". In: *Applied Intelligence* 52.8 (June 2022). Num Pages: 8337-8350 Place: Boston, Netherlands Publisher: Springer Nature B.V., pp. 8337–8350.

[12] Brendan J. Frey et al. "Clustering by Passing Messages Between Data Points". In: *Science* 315.5814 (Feb. 16, 2007). Publisher: American Association for the Advancement of Science, pp. 972–976.

[13] Arthur S. Reber. "Implicit learning of artificial grammars". In: *Journal of Verbal Learning and Verbal Behavior* 6.6 (Dec. 1, 1967), pp. 855–863.

[14] Wei Xu et al. "Detecting large-scale system problems by mining console logs". In: *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*. SOSP '09. New York, NY, USA: Association for Computing Machinery, Oct. 11, 2009, pp. 117–132.

[15] Jieming Zhu et al. "Loghub: A Large Collection of System Log Datasets for AI-driven Log Analytics". In: *2023 IEEE 34th International Symposium on Software Reliability*

Engineering (ISSRE). 2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE). ISSN: 2332-6549. Oct. 2023, pp. 355–366.

Appendix A IMV Score Visualization

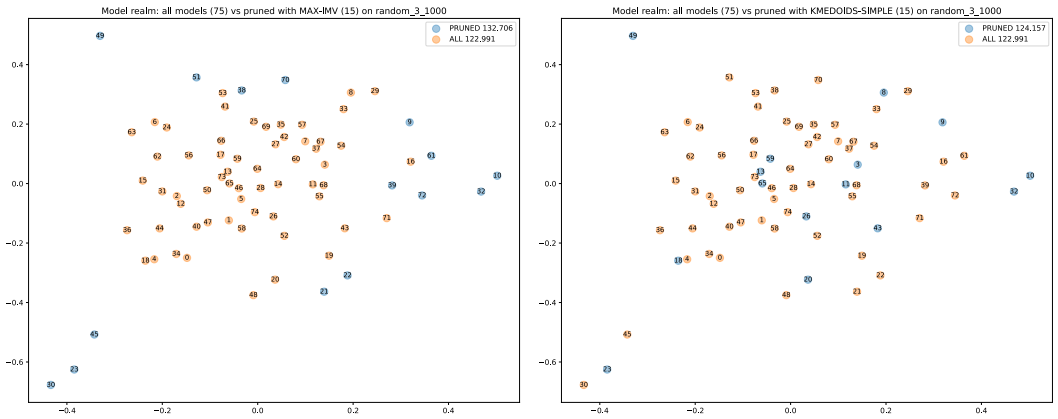

Fig. 3. Spatial visualization of models selected by pruning a bigger ensemble with Max-IMV (on the left) and K-Medoids (on the right). The plots depict the difference in the behavior of the two pruning methods. The visualization was done by reducing a distance matrix formed with the custom distance metric using Multidimensional Scaling.

Appendix B Code Implementation

Most of the key Alergia ensemble implementation was built on top of the FlexFringe [2] project. The functionality added to the FlexFringe's C++ codebase includes:

- Run configurations to train ensembles and use them to predict trace probabilities.
- Computation of the perplexity score using a provided file with a set of target probabilities.
- Computation of the pairwise sample cross entropy values for ensemble models.
- Option to set ensemble weights for predicting traces.
- Option to define a pruned subset of models that should be used for predicting traces.

All the remaining parts of the implementation such as the various pruning techniques were all implemented as complementary Python code. The Python codebase also includes all of the code to orchestrate FlexFringe runs and experiments. The utility code used for running FlexFringe with various configurations can be found in `ffutils.py`. The implementation of the custom distance metric, the IMV score and all pruning methods is contained in `variety.py`. PDFA random machines, generation of test and train sets can be found in `pdfa.py`. The complementary code also includes definitions of the PDFAs used in the experiments such as `reber`, `random_1`, etc...

All the code used for the research is publicly available in the following repositories:

- Fork of FlexFringe: `https://github.com/BlazuLyda/FlexFringeEnsemble`.
- Python code: `https://github.com/BlazuLyda/FlexFringeUtils`.

Appendix C Use of Generative Models

The creation of this paper was supported by the large language model ChatGPT. The use of this model was restricted to improving the style, wording and clarity of the written text, but all of the presented ideas, reasoning and conclusions belong to the author. Furthermore, the text sourced from the model was used as a support rather than a starting point in the writing process.

When it comes to the use of the ChatGPT model in the research process, it was used to accelerate repeatable tasks such as writing the code for plotting the graphs. Furthermore, the model was also used as a search engine for accessing the academic literature. All of the sources recommended by the model were personally checked for quality and applicability to the research by the author.