Shapley Consensus Deep Learning for Ensemble Pruning
Supplementary Materials

1. Computing the Shapley value for the function $avgmin$.

Let us assume that each player i has an intrinsic value x_i. The characteristic function is: $v(C) = avg\min_{i \in C} x_i$. Or, we can write it as follows:

$$v(C) = \frac{\sum_{i \in C} \min_{i \in C} x_i}{|C|}$$

. Without the loss of generality, let us assume that the players are indexed according to increasing x_i, i.e., $x_i <= x_j$ for $i < j$. Here n is the number of all the players in the game.

Theorem 1 Let each $i \in N$ have an intrinsic value x_i and let the characteristic function be defined as follows $v(C) = avg\min_{i \in C} x_i$. Then the Shapley value is: $SV_i(v) = x_i(n-1)! + \sum_{h=2}^{n} \sum_{k=1}^{n-h+1} -\frac{k}{h}\binom{n-k-1}{h-2}$ $(h-2)!(n-h)!$.

2. The Airport Problem

In the Airport Problem, we are given a finite set of agents $[n] = \{1, \dots n\}$, and a non-negative cost function $C : [n] \to \mathbb{R}_+ \cup \{0\}$ satisfying the condition: $i \leq j \Rightarrow C(i) \leq C(j)$ for all $i, j \in [n]$. We can associate a particular type of a cooperative game, namely the so-called Airport Game, with an Airport Problem. Given the Airport Problem $([n], C)$ we define the coalition value function $c : \mathcal{P}([n]) \to \mathbb{R}_+ \cup \{0\}$ of the Airport Game as follows: $c(S) = \max_{i \in S} C(i)$, where $S \in \mathcal{P}([n])$ is an arbitrary subset of the set of agents $[n]$, i.e., an arbitrary coalition. Since the cost of every singleton coalition $c(\{i\})$ is equal to the cost $C(i)$, not only an airport problem generates an airport game, but we can also recover an airport problem from every airport game, so the two objects can be treated as equivalent.

In our framework, the players of an airport game correspond to particular models in $\mathcal{MP}$, and their costs C are simply equal to their performances on particular data points. Therefore, for each single data point, we can compute the Shapley (Banzhaf) Values of all coalitions of models from $\mathcal{MP}$, and then compute the coalition function as an average of the above over all l data points considered.

3. Banzhaf Index for the Airport Problem

In this section we present a polynomial-time algorithm for computing the Banzhaf Index for the Airport Problem, which, by the same argument as the once concerning the Shapley Value gives us a polynomial time algorithm for determining the coalition function in our model, as it is an average of coalition functions of l equivalent airport problems. To clarify the measure of complexity, when we say that it is $f(n)$-time computable to find the Banzhaf Values for the Airport Problem, we mean that there exists an algorithm such that on the input $([n], C)$, it outputs the Banzhaf Values of all players $\{1, \dots, n\}$ in time $f(n)$. Having said that, we may proceed to the main theorem and its proof.

Theorem 2 There exists a polynomial time algorithm for computing the Banzhaf Values of all agents involved in the Airport Problem $([n], C)$.

Proof 1 Let $([n], C)$ be an arbitrary instance of an airport problem, and let $C(1) \leq \dots \leq C(n)$ be the costs of the agents taking part in the game.

For agent $i = 1$ there is exactly one way in which it can have a non-zero contribution to the value of any coalition, namely when a singleton coalition $\{i\}$ is being formed out of the empty coalition $\emptyset$. This is easy to observe, as by the structure of the cost function C and by the definition of the coalition value function c this is the only possibility. Thus, trivially, there is also only one possible value of a marginal contribution of player 1 to any coalition. For every agent $k \geq 2$ in the set of players $[n]$ there are at most as many possible marginal contribution values as there are differences $C(k) - C(i)$ for $i < k$ in $[n]$, i.e., exactly $k - 1$ values, if all $C(i)$'s for $i < k$ are pairwise different, that is when the order of numbers $C(1) \leq \dots \leq C(k)$ is strict. Computation of the Banzhaf values of players in the airport game will be feasible in polynomial time thanks to the fact that by the definition of $c(S)$ as the maximum $C(i)$ over all $i \in S$ and that by the ordering of players w.r.t. the C values, we are able to directly compute the number of coalitions S to which player k makes a non-trivial contribution and we know what this contribution will exactly be for each such coalition. To see

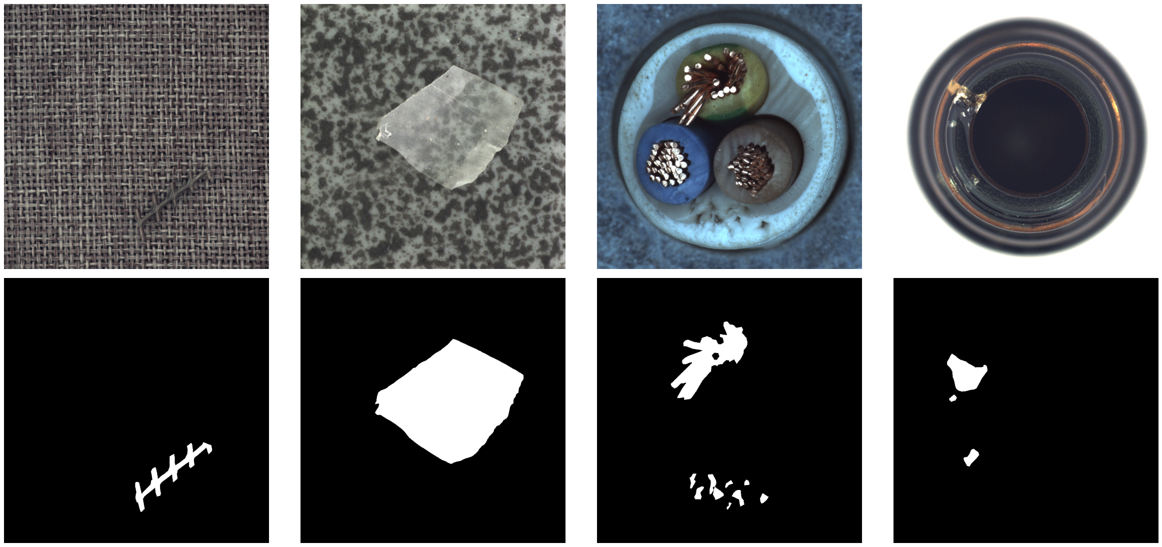

Figure 1. Some positive detection results of the SCDL where the other baseline solutions fail in the detection.

that, fix player $k \geq 2$. This agent can make the contribution $C(k) - C(i)$ for $i < k$ for those coalitions S, for which $\max(S) = i$. The number of such coalitions is exactly the number of subsets of $\{1, \ldots, i-1\}$, that is 2^{i-1}. Therefore for each $i < k$, the marginal contribution of the player k, that is, $C(k) - C(i)$ counts exactly 2^{i-1} times, if the ordering of C values is strict, which we can assume without generality. For instance, let $k = 7$, and suppose $i = 4$. To compute the Banzhaf index of player k, it is necessary (but not sufficient) to compute the number of coalitions to which player k will make the marginal contribution equal to $C(7) - C(4)$. Since $C(1) \leq C(2) \leq C(3) \leq C(4) \leq C(5)$, we need to see in how many coalitions $C(4)$ is the maximum cost value. There are exactly 2^3 such coalitions (all the coalitions corresponding to subsets of the set $\{1, 2, 3\}$) so the value $C(7) - C(4)$ enters the Banzhaf index of player 7 with the coefficient 2^3. Therefore, for each agent $k \in [n]$, its Banzhaf index is equal to:

$$\beta(k) = \frac{1}{2^{n-1}} \left[C(k) + \sum_{i=1}^{k-1} \left(2^{i-1} (C(k) - C(i)) \right) \right].$$

For $k = 1$, the sum in the bracket is simply empty, and only the cost $C(1)$ is counted into the computation of 1's Banzhaf index, which equals $\frac{C(1)}{2^{n-1}}$. Thus, in order to compute the Banzhaf power indices of all agents $k \in [n]$ we need to compute at most a sum of $k-1$ differential values $C(k) - C(i)$ for each $k \in [n]$ which requires the total of:

$$\sum_{k=1}^{n} \left(\sum_{i=1}^{k-1} i \right) = \sum_{k=1}^{n} \frac{k(k-1)}{2} = \frac{n(n-1)(n+1)}{6} = O(n^3)$$

operations.

4. More Qualitative Results

More experiments have been carried for defect detection using MVTec AD [1]. The MVTec AD challenge for anomaly detection is caused by a variety of cases of irregular defects. This dataset was introduced as a benchmark dataset for industrial anomaly detection. It contains a large variety of images depicting normal and anomalous objects in various industrial settings. Each image in the dataset is resized to 256×256. It contains $5,354$ images divided into 15 texture and object categories. This dataset covers different categories such as bottles, cans, machinery, textures, and more. Each category includes defect-free objects as well as broken, contaminated, and bent objects. Figure 1 shows some results where SCDL succeeded, and the other baseline solutions fail. From these results, we can see that SCDL is able to identify different shapes with different sizes, which is challenging for the baseline solutions. This result is achieved thanks to Shapley and the knowledge base which prunes and guides SCDL models to converge to the global optimum. Thus, the choice of the model used in the inference depends to the collaborative of the models and the similar training data.

References

[1] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019. 2