Handling Unforeseen Failures Using Argumentation-Based Learning*

H. Ayoobi[1], M. Cao[2], R. Verbrugge[1] and B. Verheij[1]

***Abstract*— General Purpose Service Robots operate in different environments of a dynamic nature. Even the robot's programmer cannot predict what kind of failure conditions a robot may confront in its lifetime. Therefore, general purpose service robots need to efficiently handle unforeseen failure conditions. This requires the capability of handling unforeseen failures while the robot is performing a task. Existing research typically offers special-purpose solutions depending on what has been foreseen at the design time. In this research, we propose a general purpose argumentation-based architecture which is able to autonomously recover from unforeseen failures. We compare the proposed method with existing incremental online learning methods in the literature. The results show that the proposed argumentation-based learning approach is capable of learning complex scenarios faster with a lower number of observations. Moreover, the final precision of the proposed method is higher than other methods.**

I. INTRODUCTION

The development and application of domestic service robots are growing rapidly. Whereas basic household robots are already common practice [1], the study of General Purpose Domestic Service Robots (GPSR) able to do complex tasks is increasing [2], [3]. Due to the dynamic environment around GPSRs, they need to efficiently handle noise and uncertainty [4].

On the architecture level of GPSRs, any kind of failure should be avoided. On a practical level, which involves persistent changes in the environment, it becomes much more difficult to account for all possible failure conditions at design time. Therefore, it is important to note that confronting unforeseen failures is mostly the default state for GPSRs, rather than an exceptional state as often described in the literature. There are some solutions for external failure recovery in the literature, which involve using simulations for the prediction of future faults [5] and logic-based reasoning to account for failures [6], [7]. However, in most of these cases, the solutions are proposed for specific applications. In this paper, we propose an argumentation-based incremental online learning method for recovering from unforeseen failures.

A. Argumentation

Argumentation is a reasoning model based on interaction between arguments [8]. In [9], Dung has defined an Abstract Argumentation Framework (AF) as the pair of the arguments (whose inner structures are unknown) and a binary relation representing the attack relation among the arguments. Extending Dung's idea, some arguments can support a conclusion and others might be against (attacking) that conclusion in the bipolar argumentation framework [10].

B. Argumentation in Machine Learning

According to a recent survey by Cocarascu et al. [11], the works using argumentation in supervised learning are listed as follows. Argumentation-Based Machine Learning (ABML) [12] uses the CN2 classification approach [13]. This method uses experts' arguments to improve the classification results. The paper by Amgoud et al. [14] explicitly uses argumentation. There are other approaches for improving classification using argumentation in the literature [15]. In contrast with the aforementioned methods, we are not using argumentation for improving the current machine learning approaches or resolving conflicting decisions between current classification methods; instead, we focus on the development of a supervised incremental learning method.

C. Research goals

We aim to develop a method with the following properties:

1) Confronting a previously unknown failure state, the robot should be able to generate related hypotheses for choosing the best recovery behavior.

2) The robot should be able to update the set of hypotheses when new contradicting facts enter the model and reason according to them.

3) The learning model should be able to learn faster than other methods with fewer number of observations.

4) The final learning precision of the model should be higher than other state-of-the-art incremental online learning methods.

II. BACKGROUND

Abstract Argumentation Framework (AF) and Bipolar Argumentation Framework (BAF) are the building blocks of the online incremental learning approach proposed in this paper. AF, BAF and online incremental machine learning algorithms are formally defined in this section.

A. Formal Definition of Abstract Argumentation Framework

An argumentation framework defined by Dung [9] is a pair $AF = (AR, attacks)$ where AR is a set of arguments, and attacks is a binary relation on AR, i.e. $attacks \subseteq AR \times AR$. The meaning of *attacks (A, B)* is that A attacks B where A and B are two arguments. In order to define the *grounded extension* semantics in AF, which is used in the proposed learning method, some semantics should be defined first.

*This work is conducted at the centre of Data Science and Systems Complexity (DSSC), and sponsored with a Marie Sklodowska-Curie COFUND grant, agreement no. 754315.

[1]Department of Artificial Intelligence, Bernoulli Institute, Faculty of Science and Engineering, University of Groningen, The Netherlands

[2] Institute of Engineering and Technology (ENTEG), Faculty of Science and Engineering, University of Groningen, The Netherlands

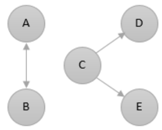

Fig. 1: An abstract argumentation framework (AF)

(**Conflict-Free**) A set S of arguments is *conflict-free* iff there are no arguments *A* and *B* in *S* such that *A* attacks *B*.

(**Acceptability**) An argument $A \in AR$ is *acceptable* with respect to a set *S* of arguments iff for each argument $B \in AR$: if *B* attacks *A* then *B* is attacked by at least one element of *S*.

(**Admissibility**) A conflict-free set of arguments S is *admissible* iff each argument in *S* is acceptable with respect to *S*.

(**Characteristic Function**) The *characteristic function* F_{AF} in an argumentation framework *AF = (AR, attacks)* is defined as follows:

$F_{AF} : 2^{AR} \rightarrow 2^{AR}$ and

$F_{AF}(S) = \{A \mid A$ is acceptable with respect to $S\}$.

(**Grounded Extension**) The *grounded extension* of an argumentation framework *AF*, denoted by GE_{AF} is the least fixed point of F_{AF}.

Example: Consider the argument set AR = {A,B,C,D,E} and the attack relations given by $attack = \{(A, B), (B, A), (C, D), (C, E)\}$ as demonstrated in Figure 1. Then the conflict-free sets of arguments would be {}, {A}, {B}, {C}, {D}, {E},{A, C}, {A, D}, {A, E}, {B, C}, {B, D}, {B, E}, {D, E}, {A, D, E}, {B, D, E}. Among these, only the sets of {}, {A}, {B}, {C}, {A, C}, {B, C} are admissible. The grounded extension is {C}.

B. Formal Definition of an Abstract Bipolar Argumentation Framework

An Abstract Bipolar Argumentation Framework (*BAF*) is an extension of Abstract Argumentation Framework by adding a support relationship. A *BAF* is a triple of the form $< AR, R_{att}, R_{sup} >$ where *AR* is the finite set of arguments, $R_{att} \subseteq AR \times AR$ is the *attack* set and $R_{sup} \subseteq AR \times AR$ is the *support* set. Considering A_i and $A_j \in AR$, then $A_i\ R_{att}\ A_j$ means that A_i attacks A_j and $A_i\ R_{sup}\ A_j$ means that A_i supports the argument A_j.

The semantics of BAF are as follows:

(**Conflict-Free**) Let $S \subseteq AR$. S is conflict-free iff $\nexists B, C \in S$ such that B attacks C.

(**Admissible set**) Let $S \subseteq AR$. S is admissible iff S is conflict-free, closed for R_{sup} (if $B \in S$ and $B\ R_{sup}\ C \Rightarrow C \in S$) and defends all its elements.

(**Preferred extension**) A set $E \subseteq AR$ is a preferred extension iff E is inclusion-maximal among the admissible sets.

Figure 2 shows a bipolar argumentation framework. Here the $\nrightarrow$ arrows show attack relations and the $\rightarrow$ arrows demonstrate support relations. The admissible sets are {}, {E}, {A, C, E}, {A, C, E, F}. The preferred extension in this BAF is {A, C, E, F}.

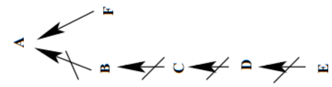

Fig. 2: An bipolar argumentation framework (BAF).

C. Formal Definition of On-line Incremental Machine Learning Algorithms

We define an incremental learning approach that uses a sequence of data instances $d_1, d_2, ..., d_t$ for generating the corresponding models $M_1, M_2, ..., M_t$. In case of supervised incremental online learning, each data instance has a label $d_i = (x_i, y_i) \in \mathbb{R}^n \times \{1, ..., C\}$ and $M_i : \mathbb{R}^n \rightarrow \{1, ..., C\}$ is representing the model which depends on M_{i-1}. The on-line learning is then defined as an incremental learning which is also able to continuously learn. Incremental learning approaches has the following properties:

- The model should adapt gradually, i.e. M_i is updated using M_{i-1}.
- The previously learned knowledge should be preserved.

III. Scenarios

The performance of the different methods is tested using a test scenario. The aim of the scenario is to model a situation where a programmer has provided an initial solution (e.g., a top level behavior such as entering the room), while he did not account for all possible failures (e.g., objects and persons blocking the entrance), but allows the robot to find new solutions whenever a (previously unseen) failure occurs.

The basic setup of the scenario is illustrated in Figure 3. The high level behavior of the robot aims to proceed from the initial location to the target location using three entrances. Different obstacles may be present in its way to the target location.

In this paper, we only concentrate on finding the best recovery solution for each failure state.

A. Recovery Solutions

Whenever the robot is confronted with a failure state, it may use any of the following recovery solutions to resolve the issue:

- Continue: This solution is only useful if the failure has resolved itself (e.g., the obstacle moved away just after the failure).
- Push: The robot can try pushing any obstacle.
- Ask: The robot can try to ask any type.
- Alternative Route: The robot can move to another entrance to reach the target location.

Notice that choosing Alternative Route as the best recovery behavior may not always lead to success, because the robot may again confront with obstacles (Figure 3). Moreover, the best recovery solution not only depends on the type of obstacle, but also on the color and location of the obstacle.

B. Test Scenario

In this scenario, three types of obstacles (ball, box or person) with four colors (red, blue, green or yellow) can be presented in one of the locations 1 to 6 (Figure 3). There can be either zero or one combination of color-type in each location.

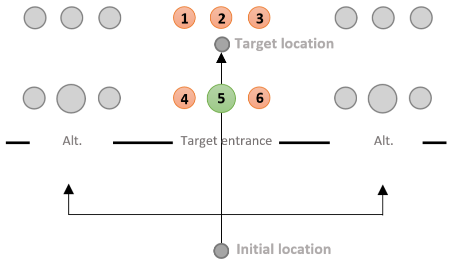

Fig. 3: Schematic overview of the possible failure state scenario. Only the green location is relevant for finding the best recovery behavior. Alt. stands for Alternative Route recovery behavior.

Only the location number 5, marked in green (Figure 3), is relevant for choosing the best recovery behavior. The number of possible combinations of color-type in each location is 13 (3 types $\times$ 4 colors + “no obstacle” = 13). Since there are 6 locations in this scenario, the number of all possible states is $13^6 = 4,826,809$.
This scenario is inspired by Ron Snijders' master thesis (University of Groningen, 2016).

IV. Method

In this section, we will talk about the proposed argumentation-based learning method for recovering from an unforeseen failure state.

A. Argumentation-Based Learning (ABL)

In order to explain ABL, we first use a simplified version of the previous test scenario where there is only one location ahead of the robot (instead of 6). When there is no obstacle ahead of the robot, the best recovery behavior is “Continue”.

Assume that the robot confronts a blue-ball blocking the entrance. Since there is no pre-trained model yet, the robot tests different recovery behaviors to find the best one (assuming that only one is successful and others will fail). Supposing that pushing the ball was successful in this case, the robot should learn from this experience.

However, only learning the recovery solutions for exactly the same experiences is not enough. We need a learning approach capable of inferring the features or feature values which are the reason for choosing the best recovery behavior. This is known as *generalization* in machine learning literature. For instance, confronting a red-ball and a green-ball with the same recovery solution of pushing, the robot should make a new hypothesis *push the ball*. Therefore, the next time the robot confronts the green-ball, it can easily infer that *Push* is the best recovery behavior.

Confronting a yellow-ball with *Alt* as the best recovery behavior contradicts the previous hypothesis. Therefore, a new hypothesis is made: *Push the ball unless it's yellow*. From an argumentation perspective, we can see each hypothesis as an argument. Therefore, the second generated hypothesis can attack and defeat the first argument. This is inspired by how human agents make new hypotheses from their perceptions and reason about the best course of action in each state.

The architecture of the proposed argumentative-learning method is shown in Figure 4. Bipolar argumentation framework is used as hypotheses generator unit and Abstract argumentation framework is modeling the defeasibility relation between these generated hypotheses in this architecture.

We now use an example to explain the proposed method.

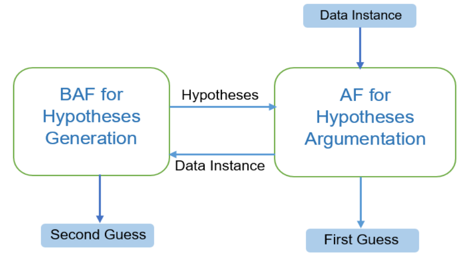

Fig. 4: Architecture of the proposed Argumentation-based learning method.

Order of Robot's Observations	*Color*	*Concept*	*Best Recovery Behavior*
1	Red	Ball	Push
2	Red	Box	Alternative route
3	Red	Person	Ask
4	Green	Ball	Push
5	Green	Box	Alternative route
6	Green	Person	Ask
7	Blue	Ball	Push
8	Blue	Box	Alternative route
9	Blue	Person	Alternative route
10	Yellow	Ball	Push
11	Yellow	Box	Alternative route
12	Yellow	Person	Ask
13	None	None	Continue

TABLE I: Possible combinations of color-type with the best recovery behaviors

B. Example

Table I shows the best recovery behavior when the robot confronts an obstacle with different colors and types. Figure 5 shows the updating procedure of the model step by step. In the hypotheses generation unit (*BAF*), an arrow $\rightarrow$ shows a support relation between arguments and $\nrightarrow$ shows an attack relation between them. However, in *AF*, $\rightarrow$ shows an attack relationship between the arguments.

Referring to Table I, at the beginning of the learning procedure, the robot confronts a Red-Ball (R-Ba). Therefore, it tests all the recovery behaviors and finds the *Push* recovery behavior as a success (Table I). Therefore, the Bipolar Argumentation Framework is getting updated as in Figure 5. In order to update the *BAF*, first, the best recovery node is constructed which is *Push* in this case. Then all the possible combinations of the feature values of the current state are added as supporting nodes. The supporting nodes are *R*, *Ba* and *R-Ba* which support the *Push* node. If there was previously the same supporting node, its supporting weight will be increased. For instance in Figure 5, where *8:B-Bo* enters the *BAF*, since *B* and *B-Ba* are new supporting nodes for the *Alt* recovery behavior, they will be added to the model with supporting weight equal to 1. On the other hand, *Bo* already exists in the set of supporting nodes for *Alt* and its weight will be increased. Confronting the *2:R-Bo* and using the previously generated hypotheses, the robot would infer that the best possible recovery behavior is *Push*, which is a wrong choice in this case (Table I). Therefore, the robot tries other recovery behaviors and finds *Alt* as success and updates the model accordingly. Moreover, a bidirectional attack will

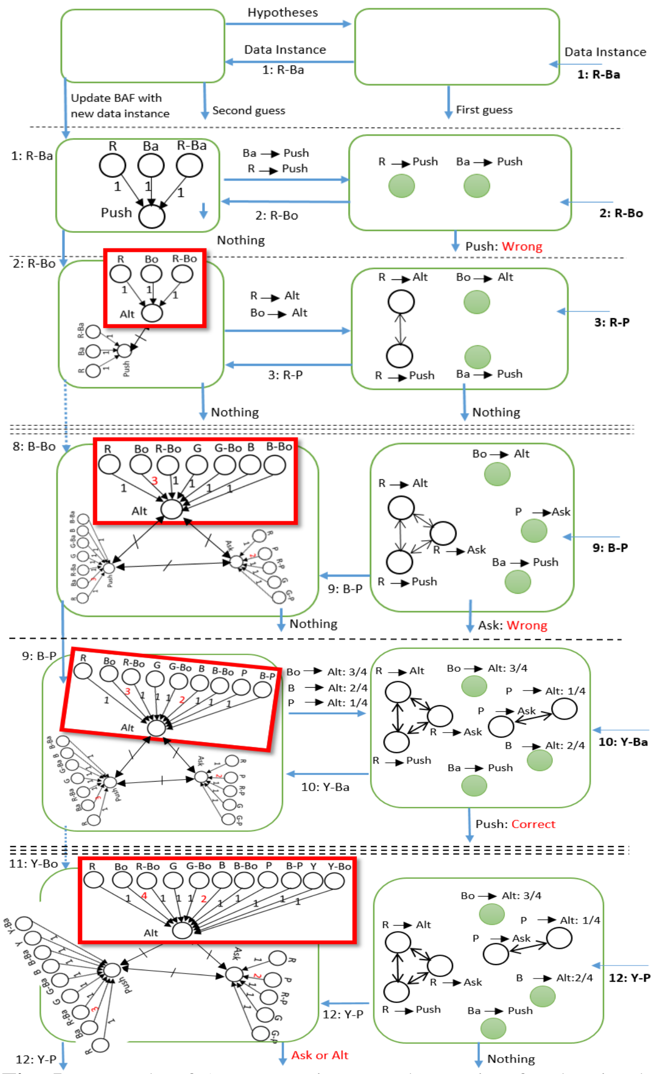

Fig. 5: Example of Argumentation-Based Learning for the simple scenario. Here only observations number 1, 2, 3, 9, 10 and 12 of Table I are shown selectively.

be added among all the recovery nodes in the *BAF* (in this case, *Alt* and *Push*). Subsequently, the new set of hypotheses is generated to update the hypotheses argumentation unit. Finally, from the set of generated hypotheses (arguments), an Abstract Argumentation Framework will be made. This *BAF-AF* update cycle goes on and on during the learning procedure.

In this tiny example, seven out of thirteen cases are correctly recovered from new failure situations and only two are wrongly classified using our proposed argumentation-based learning. In other cases, our system can provide multiple probable guesses. For instance, when *12:Y-P* enters the system in Figure 5, the *AF* cannot provide any suggestion but the *BAF* will suggest both *Ask* and *Alt* as the candidate recovery behaviors.

C. Hypotheses Generation Unit (BAF Unit)

This unit has two roles. Firstly, it generates a new set of hypotheses whenever the second unit could not classify the new data instance correctly (1). The second role of this unit is to produce a second guess for best recovery behavior (2):

Algorithm 1: Hypotheses generation pseudocode.

input: current BAF Graph, Threshold, the best recovery behavior and the latest hypothesis with wrong recovery behavior called **WrongRule**
output: set of hypotheses

- choose the Best Recovery Behavior node called **BRB**.
- Normalize the supporting weights of BRB to [0, 1].
- Sort **BRB**.supporting-nodes according to their **weight** values from high to low.

- **Sum** = 0;
- **Hypotheses-List** = Empty;

for *(any **sup** in **BRB**.supporting-nodes)* **do**
  if *(**sup**.weight $>$ **Threshold**)* **then**
    Add **sup** $\rightarrow$ **BRB** to the **Hypotheses-List**;

for *any (**A** $\rightarrow$ **BRB**) in **Hypotheses-List*** **do**
  for *any **B** $\rightarrow$ **BRB** in **Hypotheses-List*** **do**
    if *(**A** $\supset$ **B**)* **then**
      Remove (**A** $\rightarrow$ **BRB**) from **Hypotheses-List**;

Add **WrongRule**.$Precondition$ $\rightarrow$ **BRB** to Hypotheses-List;
return Hypotheses-List as output;

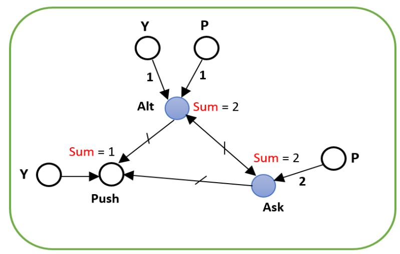

Fig. 6: The generated *BAF* when Yellow-Person (*12:Y-P*) enters the model. Blue nodes show the intersection of preferred extensions and recovery behaviors nodes.

1) In order to generate a new set of hypotheses from the constructed *BAF*, only one recovery behavior is considered which is highlighted with a red box in Figure 5. The pseudocode shown in Algorithm 1 represents how the set of hypotheses is generated.

2) In order to generate a second guess, a new *BAF* should be constructed. For the new unforeseen failure state, the set of all possible combinations of feature values is compared with the supporting nodes of each recovery behavior node. According to the sum of the matching supporting weights for any recovery behavior, the attack relationship is adapted. Therefore, only recovery behaviors with the higher sum of matching supporting weights can attack the other recovery behavior. For instance, in the example, if *12: Y-P* enters the model for prediction, the *AF* will not be able to guess the best recovery behavior. Constructing a new *BAF* for a second guess, shown in Figure 6, the calculated weight sum for *Alt* node is the same as *Ask* and higher than *Push*. Accordingly, the attack relationship gets updated. Using preferred extension semantics and its intersection with recovery behavior nodes, both *Alt* and *Ask* will be chosen as second guesses.

D. Hypotheses Argumentation Unit using AF

As stated in the previous sections, this unit tries to justify what has been learned so far by updating the attack relationship between the arguments (hypotheses). The arguments in this framework can only bidirectionally attack each other when they have the same precondition but different conclusions.

Each argument in this unit has the form *precondition* $\rightarrow$ *conclusion (postcondition) : weight*. When the new data instance enters the model, there will be three possible cases for the set of hypotheses in the grounded extension of the *AF*. Notice that in the proposed argumentation-based learning method, it can be proved that the grounded extension is a set of singletons in the *AF*. One possibility is the empty set, which leads to generating a second guess by the *BAF* unit. If one or more argument with the same conclusion part s presented, then this conclusion will be the *AF*'s first guess. If more than one argument with different recovery behaviors in their conclusion part was chosen, the weights of arguments determine which argument has more power to be selected. For instance in the example, if blue-ball enters the model after it has been trained using the complete set of data in Table-I, both *B* $\rightarrow$ *Alt: 2/4* and *Ba* $\rightarrow$ *Push:1* can be used for prediction. Since the *Ba* $\rightarrow$ *Push:1* has higher weight, the *Push* recovery behavior will be chosen, which is the correct choice for this failure state.

V. Experiments

In this section, we compare the performance of our proposed unforeseen failure recovery method with other incremental learning classifiers. Gepperth et al. compared different incremental learning algorithms and their applications [16]. Moreover, one of the recent surveys on a comparison of incremental online machine learning techniques [17], covers a broad range of algorithms. According to their results, we are also comparing the proposed method with Incremental Support Vector Machine (*ISVM*) [18], [19], [20], incremental decision tree based on C4.5 [21] and ID3, incremental Bayesian classifier [22], Online Random Forest (*ORF*)[23] and Multi-Layer Neural Networks for classification with localist models like Radial Basis Functions (RBF) which work reliably in incremental settings [24], [25].

A. Performance Measure

The mean performance of each method is calculated over 1000 independent runs. Each run consists of 200 failure recovery attempts. The order of failures is randomized for each run in which there is an equal uniform probability for each solution to be a success.

We are interested in knowing whether the method picked the best recovery solution or not for a given failure state.

Notice that all the methods use the same randomly generated data set compatible with the conditions mentioned in the test scenario.

B. Comparison criteria

We need a learning approach which can quickly learn to recover from failure states in a low number of attempts. Therefore, the increase in learning precision in a lower number of attempts is one important criterion (which we call *learning speed*) to evaluate the efficiency of the method. Furthermore, the *final learning precision* (after 200 attempts) is also an important criterion. Therefore, learning curves

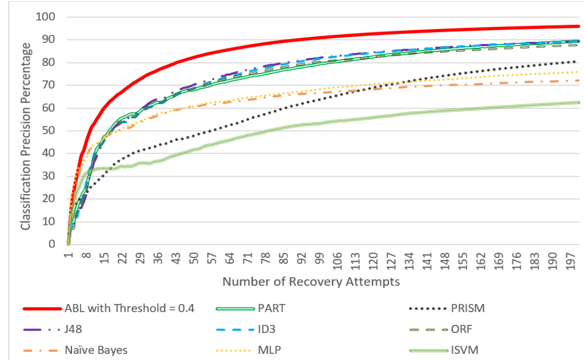

Fig. 7: The comparison of the Argumentation-Based Learning (ABL) with key methods for incremental online learning [17] using the test scenario.

with the highest steepness in a smaller number of attempts are preferred to the learning curves with almost the same steepness all over the approach that have the same final learning precision.

C. Comparison Methods

The first method utilized for comparison is a naive Bayesian classifier. The second categories of methods that are used for comparison are rule extraction and decision-tree based methods. The PART algorithm is based on the C4.5 decision tree classification method [26]. PRISM is an algorithm for inducing modular rules [27]. The ID3 algorithm constructs an unpruned decision tree for classification [28]. The J48 algorithm is also based on a pruned or unpruned C4.5 decision tree [21]. The incremental version of decision tree algorithm is discussed in [29].

The incremental version of the random forest algorithm is called On-line Random Forest (ORF) [23]. The multi-Layer Perceptron (MLP) neural network is also used for comparison. The final algorithm for the comparison is Incremental Support Vector Machine (ISVM). These are compared with our Argumentation-Based Learning (ABL) method.

D. Results

As one can see in Figure 7, the proposed Argumentation-Based Learning (ABL) method outperforms all the other methods in both the comparison criteria used for this research, namely, the final precision and the learning speed. The steepness of the learning curve shows that the ABL learns faster in a lower number of iterations. After observing 30 failure states, ABL achieves 74% precision, while the best method among others has 60% precision. The final precision of ABL is 95%, while the best final precision among other methods is 90%.

VI. Discussion

A key reason that the proposed method works better than Naive Bayes originates from the independence assumption between all features in the Naive Bayesian formulation. In the case of neural networks, since there is only a small number of training data instances, it does not work well. On the other hand, decision-tree based techniques fail at the initial recovery attempts and then gradually learn the best recovery behavior. This is because of the change in entropy or information gain when new unforeseen data updates the decision tree. This is also the case with the Online Random Forest (ORF) method. Furthermore, ISVM does not perform well in the case where only a few features are related in predicting class label. In all the above cases, the proposed ABL approach performed better since it considers any possible dependence between features and it can quickly focus on features which are most important for the classification.

Moreover, ABL leads to an explicit representation of the learning process understandable for humans, as is also the case with decision-tree based techniques. In contrast, neural networks, support vector machines and Bayesian techniques are all black boxes for human agents. This explicit representation of the learning process can be utilized in combination with human-robot interaction. By this property ABL can be used in multi-agent scenarios where agents can share their knowledge.

The proposed ABL method has limitations. It handles data sets with discrete feature values. Moreover, the complexity of the method is rather high since it uses the subsets of feature values for comparisons. These limitations can be addressed in future works.

Consequently, the proposed argumentation-based learning algorithm could learn in fewer attempts with higher precision than other algorithms used for comparison. Therefore, this method can be a good option when the feature values are discrete.

VII. Conclusion

General Purpose Service Robots should be able to recover from unexpected failure states caused by environmental changes around the robot. In this paper, an argumentation-based learning (ABL) approach is proposed which is able to generate related hypotheses (best recovery behaviors) for recovering from an unforeseen failure state. This set of hypotheses is updated incrementally and the conflicts between the hypotheses are modeled using Abstract Argumentation Frameworks. The resulting technique learns faster and with higher final precision than various state-of-the-art online incremental learning techniques.

References

[1] J. Forlizzi and C. DiSalvo, "Service robots in the domestic environment: a study of the roomba vacuum in the home," in *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction*, pp. 258–265, ACM, 2006.

[2] L. Iocchi, J. Ruiz-del Solar, and T. van der Zant, "Domestic service robots in the real world," *Journal of Intelligent & Robotic Systems*, vol. 66, no. 1, pp. 183–186, 2012.

[3] S. Schneider, F. Hegger, A. Ahmad, I. Awaad, F. Amigoni, J. Berghofer, R. Bischoff, A. Bonarini, R. Dwiputra, G. Fontana, *et al.*, "The RoCKIn@Home Challenge," in *ISR/Robotik 2014; 41st International Symposium on Robotics*, pp. 1–7, June 2014.

[4] M. J. Matarić, "Situated robotics," *Encyclopedia of Cognitive Science*, 2002.

[5] N. Akhtar, A. Kuestenmacher, P. G. Ploger, and G. Lakemeyer, "Simulation-based approach for avoiding external faults," in *16th International Conference on Advanced Robotics (ICAR)*, pp. 1–8, IEEE, 2013.

[6] A. Küstenmacher, N. Akhtar, P. G. Plöger, and G. Lakemeyer, "Unexpected situations in service robot environment: Classification and reasoning using naive physics," in *Robot Soccer World Cup*, pp. 219–230, Springer, 2013.

[7] P. Traverso, L. Spalazzi, and F. Giunchiglia, "Reasoning about acting, sensing and failure handling: a logic for agents embedded in the real world," in *International Workshop on Agent Theories, Architectures, and Languages*, pp. 65–78, Springer, 1995.

[8] F. H. Van Eemeren, B. Garssen, E. C. Krabbe, A. F. S. Henkemans, B. Verheij, and J. H. Wagemans, *Handbook of argumentation theory*. Dordrecht: Springer, 2014.

[9] P. M. Dung, "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games," *Artificial Intelligence*, vol. 77, no. 2, pp. 321–357, 1995.

[10] L. Amgoud, C. Cayrol, M.-C. Lagasquie-Schiex, and P. Livet, "On bipolarity in argumentation frameworks," *International Journal of Intelligent Systems*, vol. 23, no. 10, pp. 1062–1093, 2008.

[11] O. Cocarascu and F. Toni, "Argumentation for Machine Learning: A Survey.," in *COMMA*, pp. 219–230, 2016.

[12] M. Mozina, J. Zabkar, and I. Bratko, "Argument based machine learning," *Artificial Intelligence*, vol. 171, no. 10-15, pp. 922–937, 2007.

[13] P. Clark and T. Niblett, "The CN2 Induction Algorithm," *Machine Learning*, vol. 3, no. 4, pp. 261–283, 1989.

[14] L. Amgoud and M. Serrurier, "Agents that argue and explain classifications," *Autonomous Agents and Multi-Agent Systems*, vol. 16, no. 2, pp. 187–209, 2008.

[15] L. Carstens and F. Toni, "Using argumentation to improve classification in natural language problems," *ACM Transactions on Internet Technology (TOIT)*, vol. 17, no. 3, p. 30, 2017.

[16] A. Gepperth and B. Hammer, "Incremental learning algorithms and applications," in *European symposium on artificial neural networks (ESANN)*, 2016.

[17] V. Losing, B. Hammer, and H. Wersing, "Incremental on-line learning: A review and comparison of state of the art algorithms," *Neurocomputing*, vol. 275, pp. 1261–1274, 2018.

[18] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Advances in Neural Information Processing Systems*, pp. 409–415, 2001.

[19] B. Gu, V. S. Sheng, K. Y. Tay, W. Romano, and S. Li, "Incremental support vector learning for ordinal regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 7, no. 26, pp. 1403–1416, 2015.

[20] B. Gu, V. S. Sheng, Z. Wang, D. Ho, S. Osman, and S. Li, "Incremental learning for ν-support vector regression," *Neural Networks*, vol. 67, pp. 140–150, 2015.

[21] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[22] R. Agrawal and R. Bala, "Incremental bayesian classification for multivariate normal distribution data," *Pattern Recognition Letters*, vol. 29, no. 13, pp. 1873 – 1876, 2008.

[23] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pp. 1393–1400, IEEE, 2009.

[24] P. Reiner and B. M. Wilamowski, "Efficient incremental construction of RBF networks using quasi-gradient method," *Neurocomputing*, vol. 150, pp. 349–356, 2015.

[25] J. Lu, F. Shen, and J. Zhao, "Using self-organizing incremental neural network (soinn) for radial basis function networks," in *2014 International Joint Conference on Neural Networks (IJCNN)*, pp. 2142–2148, IEEE, 2014.

[26] E. Frank and I. H. Witten, "Generating accurate rule sets without global optimization," in *ICML*, 1998.

[27] J. Cendrowska, "Prism: An algorithm for inducing modular rules," *International Journal of Man-Machine Studies*, vol. 27, no. 4, pp. 349–370, 1987.

[28] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, Mar 1986.

[29] M. Wozniak, "A hybrid decision tree training method using data streams," *Knowledge and Information Systems*, vol. 29, no. 2, pp. 335–347, 2011.