A First Step towards Explained Activity Recognition with Computational Abstract Argumentation

Xiuyi Fan[1], Huiguo Zhang[1], Cyril Leung[2], Chunyan Miao[1]

***Abstract*— Activity recognition is a key problem in multi-sensor systems. In a home-like environment, from several sensors of different types, the multi-sensor system identifies activities performed by the inhabitants. Many supervised learning techniques exist for solving this problem. In this paper, we present a novel argumentation based approach that seamlessly combines low level sensor data processing, realized with Neural Network classifiers with high level activity recognition, represented by argumentation computation. The proposed framework gives classification results comparable to pure learning based approaches with significantly reduced training time while giving argumentative explanations.**

I. INTRODUCTION

Activity recognition has been a key problem in multi-sensor system studies [1]. Situated in a smart home environment, multiple sensors, usually of different types, collect data continuously [2]. These sensors are connected to a central computation device that runs data analytic algorithms. As a result, the inhabitants' activities are monitored and identified with abnormalities detected. Many activity recognition systems have been reported in the literature with different sensor configurations (see Section VI). This paper presents a novel argumentation based approach for solving this problem. Argumentation [3], in this context, is used as a form of rule-based reasoning system. The unique advantage of the proposed approach is that, while giving results comparable to pure machine learning approaches with significantly reduced training time, it provides argumentative explanations to the classification.

To introduce argumentation into activity recognition, the key step is to develop a theoretical framework that connects low level sensor data processing with high level argumentative reasoning. In this work, we propose *Computational Abstract Argumentation (CAA)* frameworks for this purpose. CAA frameworks are not unlike the standard Abstract Argumentation (AA) [4] frameworks in that a CAA framework contains only arguments and attacks. Unlike AA, in which arguments are atomic units, in CAA, an argument captures a purposefully designed computation. The acceptability of a CAA argument not only depends on attack relations in the CAA framework containing the argument, but also its internal computation. In this way, low level data processing is "packaged" into internal computation of arguments whereas high level argumentative reasoning remains as defined by standard argumentation semantics, e.g., admissibility.

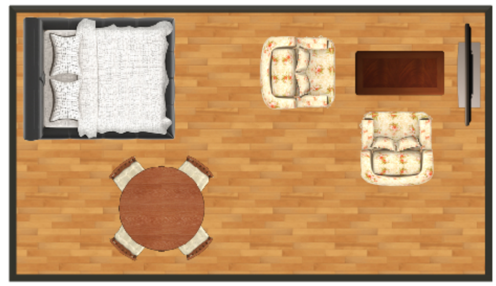

Fig. 1: Testing environment.

Multi-sensor systems differ from each other as both sensor configurations and activity recognition requirements differ between systems. Typical sensor systems include force sensors, switches, movement sensors, etc. Recognition requirements range in classification frequency (real time or not), expected accuracy, etc. Upon returning the estimated activity, it would be useful to the user if explanation for classification is provided. We leverage our previous work on argumentation based explanation [5], [6], [7].

In this work, we use a smart home environment with the following sensors:

- Two Grid-Eye Infrared Array Sensors (GridEye 1 & 2)[1],
- Two force sensors (Force 1 & 2),
- One noise sensor (Noise), and
- One electric current detector (Current).

Sensor placement is summarized in Table I with experiment environment layout illustrated in Figure 1. A GridEye thermal array sensor outputs an 8-pixel by 8-pixel thermal image in its 120-degree field of view at a 2-frame per second rate. A force sensor outputs force applied to it when there is a change to the force applied. The noise sensor measures ambient noise level once every three seconds. The current detector outputs a binary signal indicating whether the TV is on. Sample sensor outputs from one experiment session are shown in Figure 2.

Five activities we plan to recognize with our sensors are: 1) eat, 2) watch TV, 3) read books, 4) sleep, and 5) friend visit. Represented with a graph, some domain knowledge of activity classification can be described as in Figure 3. In this figure, $a1$ to $a5$ are the five activities to recognize. $s1$ to $s9$ are data processing results that are obtained from analyzing raw sensor signals. Information captured in this figure include, e.g.,

This research is supported by the National Research Foundation Singapore under its Interactive Digital Media (IDM) Strategic Research Programme.

[1]Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Nanyang Technological University, Singapore

[2]Electrical and Computer Engineering, The University of British Columbia, Canada

[1]https://na.industrial.panasonic.com/products/sensors/sensors-automotive-industrial-applications/grid-eye-infrared-array-sensor

TABLE I: Sensor placement.

GridEye 1	Ceiling above bed
GridEye 2	Ceiling above dinning table
Force 1	Sofa legs
Force 2	Surface of dinning chair
Noise	Next to TV
Current	TV's power cable

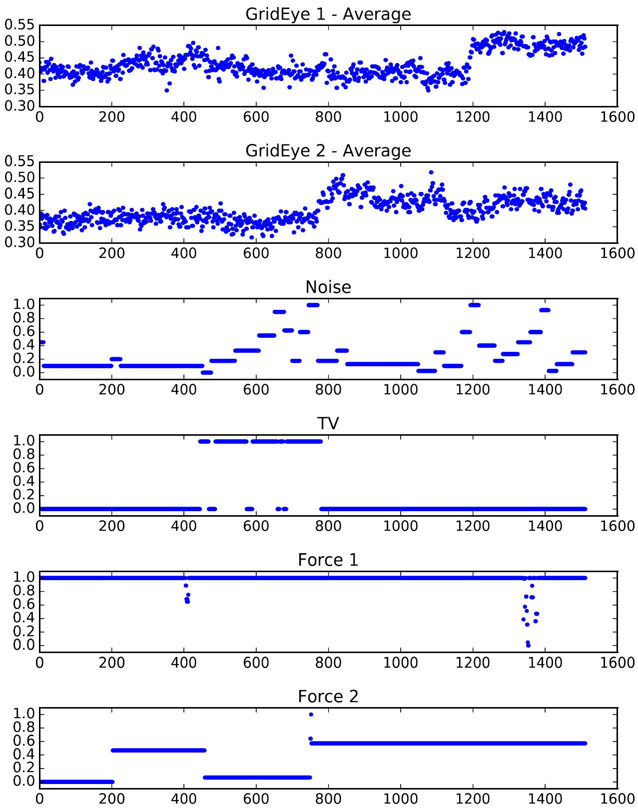

Fig. 2: Normalized data from two GridEyes (mean values), the noise sensor and the current detector during one experiment run containing all six activities (including "other"). The x-axes are time in 0.5 second resolution. The y-axes are sensor outputs.

> *if the TV is Off, then the person is not watching it* ($s4$ *connects to* $a1$).

Note that, $s1$-$s9$ are not direct sensor outputs, hence, computation mechanisms for extracting these information from sensor data are needed. The model presented in Figure 3 is by no means exhaustive or unique. It only serves the purpose of illustration. Constructions using information other than $s1$-$s9$ are entirely possible and will result in different classification performance and training time (see Section V).

The remaining of this paper is organized as follows. We introduce background on Abstract Argumentation (AA) and explanation in AA in Section II. We present Computational Abstract Argumentation in Section III. We show how explanations can be introduced in Computational Abstract Argumentation in Section IV and its applications in activity recognition. We present activity recognition experiment results in Section V. Section VI discusses some related works. We conclude in Section VII.

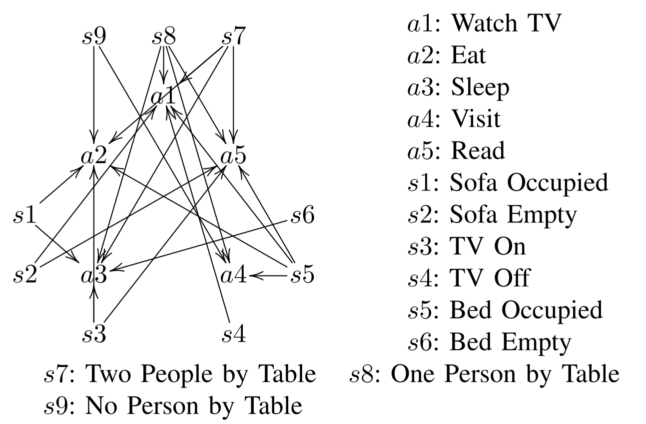

Fig. 3: Relations between sensor information and activities. This figure can be read as, for instance, *if there is no person sits next to the table, then the activity is not "Eat"* ($s9$ connects to $a2$).

II. Background

Abstract Argumentation (AA) *frameworks* [4] are pairs $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, consisting of a set of *arguments*, $\mathcal{A}$, and a binary *attack* relation, $\mathcal{R}$. Given an AA framework $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, an *extension* $A \subseteq \mathcal{A}$ is *admissible* (in *AF*) iff $\forall a, b \in A$, there is no $(a, b) \in \mathcal{R}$ (A is *conflict-free*) and for any $a \in A$, if $(c, a) \in \mathcal{R}$, then there exists some $b \in A$ such that $(b, c) \in \mathcal{R}$.

Given an AA framework $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, we say that an argument a is *in AF* iff $a \in \mathcal{A}$; we also say that an attack (a, b) is *in AF* or that a *attacks* b in *AF* iff $(a, b) \in \mathcal{R}$. We say that an argument a is *admissible* iff a is in some admissible extension.

Explanations (for non-admissible arguments) in AA [7] are defined using the *pruning operator*, $\setminus$. Given an AA framework $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ and a set of argument $A \subseteq \mathcal{A}$, $AF \setminus A = \langle \mathcal{A}', \mathcal{R}' \rangle$, where $\mathcal{A}' = \mathcal{A} \setminus A$, and $\mathcal{R}' = \{(x, y) | (x, y) \in \mathcal{R} \text{ and } x \in \mathcal{A}', y \in \mathcal{A}'\}$.

Given an AA framework $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, let $a \in \mathcal{A}$ be such that a is not admissible in *AF*. Then, if there exists some $A \subseteq \mathcal{A}$, such that: (1) a is admissible in $AF \setminus A$, and (2) there is no $A' \subset A$ such that a is admissible in $AF \setminus A'$, then A is an *explanation* of $a1$. If no such A exist in *AF*, then $\{a1\}$ is the *explanation* of $a1$.

III. Computational Abstract Argumentation

We first give the definition of Computational Abstract Argumentation (CAA) arguments.

Definition 1: A *CAA argument* is a tuple $a = \langle \mathsf{T}, \mathsf{C}, \mathsf{E} \rangle$, in which

- $\mathsf{T} \subseteq \mathtt{D}_a$: T is the *Data*, and $\mathtt{D}_a$ is the *Domain*;
- $\mathsf{C} : \mathtt{D}_a \mapsto \mathtt{R}_a$: C is the *Computation Function (Computation)*, and $\mathtt{R}_a$ is the *Range*;
- $\mathsf{E} : \mathtt{R}_a \mapsto \{\top, \bot\}$: E is the *Evaluation Function (Evaluation)*.

Informally, we can read a CAA argument $a = \langle \mathsf{T}, \mathsf{C}, \mathsf{E} \rangle$ as follows.

- the *data* T is the input for a's computation; this data can be expressed in various forms, we say that T is in some *domain* $\mathtt{D}_a$;
- the *computation* C is a function from $\mathtt{D}_a$ to $\mathtt{R}_a$ defining how data T is used, the results of this computation is in some *range* $\mathtt{R}_a$;
- the *evaluation* E describes how results obtained via C should be interpreted, it gives an binary evaluation to the *"applicability"* of a.

Overall, a CAA argument $a = \langle \mathsf{T}, \mathsf{C}, \mathsf{E} \rangle$ can be read as: given some data presented in T and some way C of processing the data, a is deemed to be "applicable" wrt. T and C, iff E returns $\top$. Argument applicability is formalized as follows.

Definition 2: A CAA argument $\langle \mathsf{T}, \mathsf{C}, \mathsf{E} \rangle$ is *applicable* iff $\mathsf{E}(\mathsf{C}(\mathsf{T})) = \top$. We say that CAA arguments that are not applicable *non-applicable*.

For any CAA argument a of the form $\langle x, (x, x), (x, \top) \rangle$ with variable x, we refer to a as a *constant argument*.[2] Intuitively, a constant argument represents its data and is always applicable.

With CAA arguments defined, we define CAA frameworks as follows.

Definition 3: A CAA Framework is a pair $\langle \mathcal{A}_c, \mathcal{R}_c \rangle$, in which $\mathcal{A}_c$ is a set of *CAA arguments* and $\mathcal{R}_c$ is a set of attack relations over $\mathcal{A}_c$.

Example 1: We instantiate a CAA framework with our activity recognition problem described in Figure 3 with nodes and arrows representing CAA arguments and attacks, respectively. The CAA arguments are follows.

- Let $a1 - a5$ be constant arguments;
- Let $s1$ / $s2$ be $\langle \mathsf{T}_{s1}, \mathsf{C}_{s1}, \mathsf{E}_{s1} \rangle$ / $\langle \mathsf{T}_{s2}, \mathsf{C}_{s2}, \mathsf{E}_{s2} \rangle$ in which
  - $x_F = \mathsf{T}_{s1} = \mathsf{T}_{s2} \subseteq \mathbb{N}$ is the output from Force 1;
  - $\mathsf{C}_{s1}(x_F) = \mathsf{C}_{s2}(x_F) = x_F$; and
  - E_{s1} / $\mathsf{E}_{s2} = \top$ if $\mathsf{C}_{s1} \leq \theta$ / $\mathsf{C}_{s2} \geq \theta$ for some threshold θ and E_{s1} / $\mathsf{E}_{s2} = \bot$ otherwise.
- Let $s3$ / $s4$ be $\langle \mathsf{T}_{s3}, \mathsf{C}_{s3}, \mathsf{E}_{s3} \rangle$ / $\langle \mathsf{T}_{s4}, \mathsf{C}_{s4}, \mathsf{E}_{s4} \rangle$ in which
  - $x_C = \mathsf{T}_{s3} = \mathsf{T}_{s4} \subseteq \{0, 1\}$ is the output from the Current sensor;
  - $\mathsf{C}_{s3}(x_C) = \mathsf{C}_{s4}(x_C) = x_C$; and
  - E_{s3} / $\mathsf{E}_{s4} = \top$ if $\mathsf{C}_{s3} = 1$ / $\mathsf{C}_{s4} = 0$ and E_{s3} / $\mathsf{E}_{s4} = \bot$ otherwise.
- Let $s5$ / $s6$ be $\langle \mathsf{T}_{s5}, \mathsf{C}_{s5}, \mathsf{E}_{s5} \rangle$ / $\langle \mathsf{T}_{s6}, \mathsf{C}_{s6}, \mathsf{E}_{s6} \rangle$ in which
  - $x_{G1} = \mathsf{T}_{s5} = \mathsf{T}_{s6} \subseteq \mathbb{N}^{8\times 8}$ is the output from GridEye 1;[3]
  - $\mathsf{C}_{s5}(x_{G1}) = \mathsf{C}_{s6}(x_{G1}) = CL_b(x_{G1}, \omega_b)$ is a classifier such that with some parameter ω_b, $CL_b(x_{G1}, \omega_b)$ is the number of people in bed.
  - E_{s5} / $\mathsf{E}_{s6} = \top$ if $\mathsf{C}_{s5} = 1$ / $\mathsf{C}_{s6} = 0$ and E_{s5} / $\mathsf{E}_{s6} = \bot$ otherwise.
- Let $s7$ / $s8$ / $s9$ be $\langle \mathsf{T}_{s7}, \mathsf{C}_{s7}, \mathsf{E}_{s7} \rangle$ / $\langle \mathsf{T}_{s8}, \mathsf{C}_{s8}, \mathsf{E}_{s8} \rangle$ / $\langle \mathsf{T}_{s9}, \mathsf{C}_{s9}, \mathsf{E}_{s9} \rangle$ in which
  - $x_{G2} = \mathsf{T}_{s7} = \mathsf{T}_{s8} = \mathsf{T}_{s9} \subseteq \mathbb{N}^{8\times 8}$ is the output from GridEye 2;
  - $\mathsf{C}_{s7}(x_{G2}) = \mathsf{C}_{s8}(x_{G2}) = \mathsf{C}_{s9}(x_{G2}) = CL_d(x_{G2}, \omega_d)$ is a classifier such that with some parameter ω_d, $CL_d(x_{G2}, \omega_d)$ is the number of people sit by the table.
  - E_{s7} / E_{s8} / $\mathsf{E}_{s9} = \top$ if $\mathsf{C}_{s7} = 2$ / $\mathsf{C}_{s8} = 1$ / $\mathsf{C}_{s9} = 0$ and E_{s7} / E_{s8} / $\mathsf{E}_{s9} = \bot$ otherwise.

We represent the five activities to be classified with five corresponding constant arguments ($a1$ - $a5$) and connect sensor outputs to the remaining arguments ($s1$ - $s8$), namely:

- $s1$ and $s2$ process data outputs from Force 1. $s1$ is applicable iff Force 1 outputs values exceed some threshold. $s2$ is applicable iff Force 1 outputs values below this threshold.
- $s3$ and $s4$ are linked to the current detector. $s3$ is applicable iff the current detector outputs 1. $s4$ is applicable iff the current detector outputs 0.
- $s5$ and $s6$ are linked to GridEye 1. $s5$ is applicable iff GridEye 1 detects enough heat to believe there is a person in bed; otherwise, $s6$ is applicable.
- $s7 - s8$ are linked to GridEye 2. The applicability of them is determined by the number of people sitting by the dinning table.

In this way, with a set of sensor output, some of the CAA arguments $s1$ - $s9$ will be applicable whereas others being non-applicable. Consequently, some of $a1$ - $a5$ will be admissible (see below). If there is one and only one CAA argument with $a1$ - $a5$ admissible, then it is the classified activity, with respect to the given set of sensor output. If none of $a1$ - $a5$ is admissible or more than one of $a1 - a5$ are admissible, then no activity is recognized.

Note that since our definition of CAA arguments, especially their computation and evaluation functions, are generic, not all CAA frameworks are suitable for problem representation and computation. We introduce *Well-formed Computational Abstract Argumentation (WCAA)* identifying the computable subset.

Definition 4: A *Well-formed Computational Abstract Argumentation (WCAA) framework* is a CAA framework $\langle \mathsf{T}, \mathsf{C}, \mathsf{E} \rangle$ such that both C and E are total and computable functions.

By letting computation C and evaluation E be total and computable, we ensure a WCAA is well defined as its computation always terminates. In the context of our activity recognition problem, we need to enforce the two classifier functions CL_b and CL_d to be total and computable.

We refer to WCAA in all subsequent discussion.

Standard argumentation semantics can be reused with minimum changes in CAA. We illustrate this with admissibility.

Definition 5: Given a WCAA framework $AF = \langle \mathcal{A}_c, \mathcal{R}_c \rangle$, a set of arguments $As \subseteq \mathcal{A}_c$ is *admissible* iff

- for all $a \in As$, a is applicable; and
- for all $a, b \in As$, $\langle a, b \rangle \notin \mathcal{R}_c$; and
- for all $a \in \mathcal{A}_c$, if a is applicable and there is some $b \in As$, such that $\langle a, b \rangle \in \mathcal{R}_c$, then there exists some $c \in As$ and $\langle c, a \rangle \in \mathcal{R}_c$.

[2] Note that (x, x) is an identity function and $(x, \top)$ is a constant function.

[3] The output of a GridEye infrared array sensor is represented by 8-by-8 integer matrices.

We say that As is an *admissible set* in AF. For all $a \in As$, we say that a is *admissible* in AF.

Directly from Definition 5, an empty set is admissible in any CAA framework. Thus, the next proposition holds.

Proposition 1: Every CAA framework possesses at least one admissible set of arguments.

It is trivial to see that constant arguments are a convenient tool to model AA's abstract arguments in CAA. Thus, the following lemma holds.

Lemma 1: AA is an instance of CAA.

Proof: To show that AA is an instance of CAA is to show that for every AA framework *AF*, there exists a CAA framework AF_c such that:

1) for every argument a in *AF*, there is a counterpart of a, a_c in AF_c;
2) a is acceptable (wrt. some semantics S) in *AF* iff a_c is acceptable in AF_c (wrt. S).

Thus, given an AA framework $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ we construct a CAA framework $AF_c = \langle \mathcal{A}_c, \mathcal{R}_c \rangle$, such that:

- for each AA argument a in *AF*, we create a CAA constant argument $a_c = \langle \{a\}, (\{a\}, \{a\}), (\{a\}, \top) \rangle$ in AF_c and there is no other CAA argument in AF_c;
- for AA arguments a, b in *AF*, let a_c and b_c be counterparts of a and b, respectively, in AF_c, if a attacks b exists in $\mathcal{R}$, then a_c attacks b_c exists in $\mathcal{R}_c$ and there is no other attack in AF_c.

Clearly, with above construction, both 1 and 2 are fulfilled as there is a one-to-one mapping between arguments and attacks in *AF* and AF_c. Since all CAA arguments in AF_c are constant arguments, they are always applicable. Thus their acceptability only depends on attack relations. Therefore, if an argument a is acceptable in *AF* wrt. any semantics S, then a_c is acceptable in AF_c wrt. S. ■

Lemma 1 establishes an important connection between AA and CAA. Since existing argumentation formalism, e.g., Assumption-based Argumentation (ABA) [8], are instances of AA, they are also instances of CAA, formally:

Corollary 1: ABA is an instance of CAA.

Proof: Since ABA is an instance of AA [9] and Lemma 1 shows that AA is an instance of CAA, this corollary holds. ■

Lemma 1 and Corollary 1 help to put CAA in to the wider context of argumentation formalism sanctioning that CAA is a generalization of some existing formalism.

IV. Explanation in CAA

With Lemma 1 showing that AA is an instance of CAA, we can use the same intuition of AA explanations in CAA. Namely, for some non-admissible CAA argument a, its non-admissibility can either be attribute to its own non-applicability, or the existences of some indefensible attackers. We formalize explanations for non-acceptable CAA arguments as follows. (Note that, we do not redefine the pruning operator, $\backslash$, for CAA frameworks but assuming its AA definition, as given in Section II applies.)

Definition 6: Given a CAA framework $AF = \langle \mathcal{A}_c, \mathcal{R}_c \rangle$, let $a \in \mathcal{A}_c$ be such that a is not admissible in *AF*. Then, if there exists some $A \subseteq \mathcal{A}_c$, such that:

1) a is admissible in $AF \setminus A$, and
2) there is no $A' \subset A$ such that a is admissible in $AF \setminus A'$, then A is an *explanation* of a.

Otherwise, $\{a\}$ is the *explanation* of a.

As in AA, explanation of a CAA argument a is defined as the minimum set of arguments A, such that, by removing A, a becomes admissible. However, such A may not always exist. In such cases, the reason for a being not admissible is a itself. It is easy to see that this happens iff a attacks itself or when a is not applicable. Formally:

Proposition 2: Given a CAA framework $AF = \langle \mathcal{A}_c, \mathcal{R}_c \rangle$, let $a \in \mathcal{A}_c$. The explanation of a is $\{a\}$ iff (1) $(a, a) \in \mathcal{R}$ or (2) a is not applicable.

Note that, intuitively, non-applicable argument cannot be in an explanation of any other argument except itself. Formally:

Proposition 3: Given a CAA framework $AF = \langle \mathcal{A}_c, \mathcal{R}_c \rangle$, let $a \in \mathcal{A}_c$ be some non-applicable argument, then there does not exist $b \neq a \in \mathcal{A}_c$ such that $a \in Bs$, in which Bs is an explanation of b.

Proof: (Sketch.) By Definition 6, an explanation Bs of any argument b needs to be the minimum set that, if removed, then b becomes admissible. By Definition 5, we know that a non-applicable argument does not affect the admissibility of any other argument. Thus, a non-applicable argument does not belong to any explanation. ■

We illustrate explanations for CAA argument with the following example.

Example 2: (Example 1 continued.) Suppose that with one particular set of sensor data, we notice that among $s2$, $s8$, $s7$ and $s5$, both $s2$, $s8$ and $s7$ are not applicable but $s5$ is. It is easy to see that in this case a is not admissible and its explanation is $s5$. Thus, the explanation of activity not being "Watch TV" is "the bed is occupied".

V. Activity Recognition Experiment

We experiment CAA based activity recognition in a simulated home-like environment as described in Section I. For data collection, we perform eight runs of experiments by four individuals with each person performing the five defined activities twice. In our experiments, each activity lasts two to three minutes in every run. For eating, the testing subject sits by the dinning table and consumes a snack. For watching TV and reading book, he sits on the same sofa with TV on and off, respectively. For sleeping, the testing subject lays on the bed with little movement. For friend visit, an additional testing subject enters the room and both sit at the dinning table carrying out a conversation.

All activities are manually labeled. For algorithm comparison, data collected from six runs from three testing subjects are used for training with the remaining two runs from the fourth testing subject for testing. Overall, there are 7781 instances of training samples and 1437 instances of testing samples collected.

With the CAA framework as given in Example 1, to perform activity recognition we need to

1) identify the threshold, θ, for the Force 1 sensor output;
2) construct the function CL_b with parameter ω_b;
3) construct the function CL_d with parameter ω_d; and
4) compute admissible argument representing activities.

With the collected training data, we let θ be the median value of all readings from Force 1. To construct the two classifiers, CL_b and CL_d we have chosen perceptrons with the standard Back Propagation training [10] for their simplicity. To test the admissibility of an argument, we have developed an implementation based on dispute trees [9].

Results from the implemented CAA-based classifier are summarized in Table II, shown in form of a confusion matrix. The table should be understood as: 201 instances of "Read" are (correctly) classified as "Read"; 13 instances of "Read" are (incorrectly) classified as "Sleep"; 46 instance of "WatchTV" are (incorrectly) classified as "Read", etc. We observe the classifier misclassifies several "WatchTV" instances as "Read". We believe these errors are results of noise from the TV current sensor.

To put our CAA-based classifier into perspective, we have also implemented several other widely used classifiers, including Naive Bayes, Decision Tree, Perceptron and Deep Neural Networks (with four hidden layers) [10]. We compare their performance using average precision, average recall and training time.[4]. The results are summarized in Table III. We can see that the CAA-based classifier performs as good as the best performer, Deep Neural Networks, but only requires a small fraction of training time. This result illustrates the effectiveness of introducing domain knowledge in solving activity recognition problems.

Explanations in the form of Example 2, e.g.,

> *"it is not Read as the TV is on"* or *"it is not Eat as the Bed is occupied"*

have been produced for our experiments. We observe that: (1) comparing with other "black-box" methods, e.g., Neural Networks and Naive Bayes, explanations given by the proposed method clearly help users to better understand the classification; and (2) comparing with Decision Tree, the proposed method also shows advantage as decision trees use individual sensor features directly, e.g., "*branch left with a particular node if pixel (2,3) of GridEye 1 is High*", yielding trees that are not only unsuitable for classification but also incapable of delivering meaningful explanation.

VI. Related Work

We review related work in two areas: firstly, works on activity recognition, and then, works on combination of logic-based inference and learning systems.

Activity recognition has been a long standing problem in multi-sensor system studies. [11] is an early work on activity recognition in home-like environment. They use a large amount of low cost state-change sensors, e.g., switches and movement sensors. For classification, they use a Naive Bayes classifier. [12], [13], [14] are works on activity recognition based on temporal pattern representation of abstract sensors models. In those works, each activity is represented as a sequence of sensor events. [13] introduces an ontology describing activities. [14] uses Hidden Markov Model and Conditional Random Fields. [15] present an abnormal behavior detection work in a home-like environment similar to ours. In that work, no explicit activity recognition is reported, only a binary classification on behavioral abnormality with recurrent neural networks (RNN). [16] is similar to [15] in that it detects abnormal behaviors. Instead of using RNN, they use a Bayesian model based approach hypothesizing "normal" sensor event patterns. [17], [18] both use wearable devices for activity recognition. [17] uses statistical calculation whereas [18] uses Gaussian Mixture Models with finite state machines. [19] experiments several deep learning models. Compare with those works, our work presented in this paper is the first one that utilizes argumentation theory.

[4]Precision is defined as True Positive / (True Positive + False Positive); recall is defined as True Positive / (True Positive + False Negative).

TABLE II: Classification Results from CAA-based Classifier.

	Read	Visit	Sleep	WatchTV	Eat
Read	201	0	13	0	0
Visit	0	319	26	0	2
Sleep	0	2	299	17	6
WatchTV	46	0	4	270	0
Eat	0	6	13	0	206

TABLE III: Performance comparison with other classifiers.

	Precision	Recall	Training Time (s)
Naive Bayes	0.682	0.574	0.86
Decision Tree	0.724	0.357	0.43
Perceptron	0.752	0.745	17.42
Deep NN	0.814	0.825	165.34
CAA-Classifier	0.822	0.824	7.43

The combination of logic-based inference and learning systems have attracted some research attention. [20] surveys some work in creating mappings between various logic formalism and connectionist models. Many of these works are concerned about translating logic based models into neural networks. In argumentation, [21] study how value-based argumentation frameworks (VAFs) can be mapped into neural networks. [22], [23] continue that line of research to study various properties of the developed VAF to neural network mapping; and [24] is the most recent extension. The work presented in this paper is of a different intuition. Namely, no attempt has been made to map CAA frameworks into neural networks, rather, CAA is a glue between argumentation and non-logic based computation, e.g., neural networks in our example. In this way, similar to [25], argumentation and neural networks are tasked to tackle different parts of a problem with argumentation representing domain knowledge and neural networks gives superior low level data processing.

[26] and [27] give two extensions to AA. Both works consider examining an arguments' validity before considering attacks when determining the argument's acceptability.

However, neither of the two is on connecting data processing and argumentative reasoning. They are not about activity recognition. Moreover, they are also not concerned with generating explanation.

[28], [29] on using argumentation to accelerate training in reinforcement learning and [30] on using argumentation to improve the performance of Naive Bayes, Random Forest and Support Vector Machines in sentiment analysis are similar to our work in spirit in that all of these works are concerned about introducing algorithms that utilize both argumentation and some traditional machine learning techniques. Our work differ from those as, instead of applying existing argumentation frameworks in an ad-hoc fashion, our work proposes a unified framework with added explanation.

VII. Conclusion

In this work, we present a novel argumentation framework, Computational Abstract Argumentation (CAA), to seamlessly connect low level data processing with high level inference based reasoning. A CAA framework is composed of arguments and attacks in which CAA arguments are self-contained "computation units" and attacks are defined over arguments as in abstract argumentation. We experiment CAA's application in solving the activity recognition problem with promising results comparable to traditional machine learning algorithms. The advantage of the proposed CAA based approach is twofold. Firstly, used as a channel for directly incorporating domain knowledge into problem solving, CAA significantly reduces the training time required for model construction. Secondly, the argumentative structure of CAA provides the basis for generating explanations for the modeled computation.

In future, we would like to explore the following directions. Firstly, in this work, we have only developed explanations for non-acceptable CAA arguments. We would like to explore justifications for acceptable arguments in future. Secondly, we would like to explore CAA's use in other applications and its properties related to existing argumentation frameworks and semantics.

Acknowledgments

This research was supported by the National Research Foundation Singapore under its Interactive Digital Media (IDM) Strategic Research Programme.

References

[1] D. J. Cook and S. K. Das, "How smart are our environments? an updated look at the state of the art," *Pervasive and Mobile Computing*, vol. 3, no. 2, pp. 53–73, 2007.

[2] M. C. Mozer, "The neural network house: An environment that adapts to its inhabitants," in *Proc. AAAI Spring Symp. Intelligent Environments*, 1998, pp. 110–114.

[3] S. Modgil, F. Toni, F. Bex, I. Bratko, C. Chesñevar, W. Dvořák, M. Falappa, X. Fan, S. Gaggl, A. García, M. González, T. Gordon, J. Leite, M. Možina, C. Reed, G. Simari, S. Szeider, P. Torroni, and S. Woltran, "The added value of argumentation," in *Agreement Technologies*. Springer, 2013, vol. 8, pp. 357–403.

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

[5] X. Fan and F. Toni, "On computing explanation in abstract argumentation," in *Proc. ECAI*, 2014, pp. 1005–1006.

[6] ——, "On computing explanations in argumentation," in *Proc. of AAAI*, 2015, pp. 1496–1502.

[7] ——, "On computing explanations for non-acceptable arguments," in *Proc. TAFA*, 2015.

[8] F. Toni, "A tutorial on assumption-based argumentation," *Argument & Computation, Special Issue: Tutorials on Structured Argumentation*, vol. 5, no. 1, pp. 89–117, 2014.

[9] P. Dung, P. Mancarella, and F. Toni, "Computing ideal sceptical argumentation," *AIJ*, vol. 171, no. 10–15, pp. 642–674, 2007.

[10] T. M. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997.

[11] E. M. Tapia, S. S. Intille, and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," in *Proc. PERVASIVE*, 2004, pp. 158–175.

[12] X. Hong, C. D. Nugent, M. D. Mulvenna, S. Martin, S. Devlin, and J. G. Wallace, "Dynamic similarity-based activity detection and recognition within smart homes," *Int. J. Pervasive Computing and Communications*, vol. 8, no. 3, pp. 264–278, 2012.

[13] G. Okeyo, L. Chen, H. Wang, and R. Sterritt, "Dynamic sensor data segmentation for real-time knowledge-driven activity recognition," *Pervasive and Mobile Computing*, vol. 10, pp. 155–172, 2014.

[14] T. Kasteren, A. K. Noulas, G. Englebienne, and B. J. A. Kröse, "Accurate activity recognition in a home setting," in *Proc. UbiComp*, 2008, pp. 1–9.

[15] A. Lotfi, C. S. Langensiepen, S. M. Mahmoud, and M. J. Akhlaghinia, "Smart homes for the elderly dementia sufferers: identification and prediction of abnormal behaviour," *J. Ambient Intelligence and Humanized Computing*, vol. 3, no. 3, pp. 205–218, 2012.

[16] F. J. Ordóñez, P. Toledo, and A. Sanchis, "Sensor-based bayesian detection of anomalous living patterns in a home setting," *Personal and Ubiquitous Computing*, vol. 19, no. 2, pp. 259–270, 2015.

[17] M. Uddin, A. Salem, I. Nam, and T. Nadeem, "Wearable sensing framework for human activity monitoring," in *Proc. WearSys 2015*, 2015, pp. 21–26.

[18] N. F. Ince, C. Min, and A. H. Tewfik, "A feature combination approach for the detection of early morning bathroom activities with wireless sensors," in *Proc. SIGMOBILE*, 2007, pp. 61–63.

[19] X. Fan, H. Zhang, C. Leung, and C. Miao, "Comparative study of machine learning algorithms for activity recognition with data sequence in home-like environment," in *Proc. IEEE MFI*, 2016, In press.

[20] L. C. Lamb, "The grand challenges and myths of neural-symbolic computation," in *Recurrent Neural Networks - Models, Capacities, and Applications, 20.01. - 25.01.2008*, 2008.

[21] A. S. Garcez, D. M. Gabbay, and L. C. Lamb, "Value-based argumentation frameworks as neural-symbolic learning systems," *JLC*, vol. 15, no. 6, pp. 1041–1058, 2005.

[22] W. Makiguchi and H. Sawamura, "A hybrid argumentation of symbolic and neural net argumentation (part I)," in *Proc. ArgMAS*, 2007, pp. 197–215.

[23] ——, "A hybrid argumentation of symbolic and neural net argumentation (part II)," in *Proc. ArgMAS*, 2007, pp. 216–233.

[24] A. S. Garcez, D. M. Gabbay, and L. C. Lamb, "A neural cognitive model of argumentation with application to legal inference and decision making," *JAL*, vol. 12, no. 2, pp. 109–127, 2014.

[25] X. Fan, S. Liu, H. Zhang, C. Leung, and C. Miao, "Explained activity recognition with computational assumption-based argumentation," in *Proc. ECAI*, 2016, In press.

[26] N. Oren and T. J. Norman, "Semantics for evidence-based argumentation," in *Proc. COMMA*, 2008, pp. 276–284.

[27] G. Brewka and S. Woltran, "Abstract dialectical frameworks," in *Proc. KR*, 2010.

[28] Y. Gao, F. Toni, and R. Craven, "Argumentation-based reinforcement learning for robocup soccer keepaway," in *Proc. ECAI*, 2012, pp. 342–347.

[29] Y. Gao and F. Toni, "Argumentation accelerated reinforcement learning for cooperative multi-agent systems," in *Proc. ECAI*, 2014, pp. 333–338.

[30] L. Carstens and F. Toni, "Improving out-of-domain sentiment polarity classification using argumentation," in *Proc. ICDMW*, 2015, pp. 1294–1301.