TAO: Context Detection from Daily Activity Patterns Using Temporal Analysis and Ontology

SUDERSHAN BOOVARAGHAVAN, Carnegie Mellon University, United States
PRASOON PATIDAR, Carnegie Mellon University, United States
YUVRAJ AGARWAL, Carnegie Mellon University, United States

Translating fine-grained activity detection (e.g., *phone ring, talking interspersed with silence and walking*) into semantically meaningful and richer contextual information (e.g., *on a phone call for 20 minutes while exercising*) is essential towards enabling a range of healthcare and human-computer interaction applications. Prior work has proposed building ontologies or temporal analysis of activity patterns with limited success in capturing complex real-world context patterns. We present TAO, a hybrid system that leverages OWL-based ontologies and temporal clustering approaches to detect high-level contexts from human activities. TAO can characterize sequential activities that happen one after the other and activities that are interleaved or occur in parallel to detect a richer set of contexts more accurately than prior work. We evaluate TAO on real-world activity datasets (*Casas* and *Extrasensory*) and show that our system achieves, on average, 87% and 80% accuracy for context detection, respectively. We deploy and evaluate TAO in a real-world setting with eight participants using our system for three hours each, demonstrating TAO's ability to capture semantically meaningful contexts in the real world. Finally, to showcase the usefulness of contexts, we prototype wellness applications that assess productivity and stress and show that the wellness metrics calculated using contexts provided by TAO are much closer to the ground truth (on average within 1.1%), as compared to the baseline approach (on average within 30%).

CCS Concepts: • **Human-centered computing** → **Ambient intelligence**; **Ubiquitous computing**; **Ubiquitous and mobile computing systems and tools**.

Additional Key Words and Phrases: Behavioral context recognition, activity recognition, ontology, deep Learning

ACM Reference Format:
Sudershan Boovaraghavan, Prasoon Patidar, and Yuvraj Agarwal. 2023. TAO: Context Detection from Daily Activity Patterns Using Temporal Analysis and Ontology. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 3, Article 87 (September 2023), 32 pages. https://doi.org/10.1145/3610896

1 INTRODUCTION

The vision of Human Activity Recognition (HAR) approaches is to enable a better understanding of human behavior for applications in healthcare and human-computer interaction [31]. Recent advances in Internet-of-Things (IoT), pervasive ambient sensing, and richer ML-based inferences have taken us even closer to this vision [9, 10, 20, 40]. A key challenge, however, is that human activity patterns are complex in nature and often require contextual information about the activity to be useful for downstream applications. For example, a wellness application that assesses an individual's productivity requires contextual information about an activity. An activity such as talking may or may not indicate if the individual is being productive, depending on whether it is happening in a context denoting office work or a different context of having a meal. Moreover, current

Authors' addresses: Sudershan Boovaraghavan, sudershan@cmu.edu, Carnegie Mellon University, Pittsburgh, United States; Prasoon Patidar, prasoonpatidar@cmu.edu, Carnegie Mellon University, Pittsburgh, United States; Yuvraj Agarwal, yuvraj@cs.cmu.edu, Carnegie Mellon University, Pittsburgh, United States.

2474-9567/2023/9-ART87
https://doi.org/10.1145/3610896

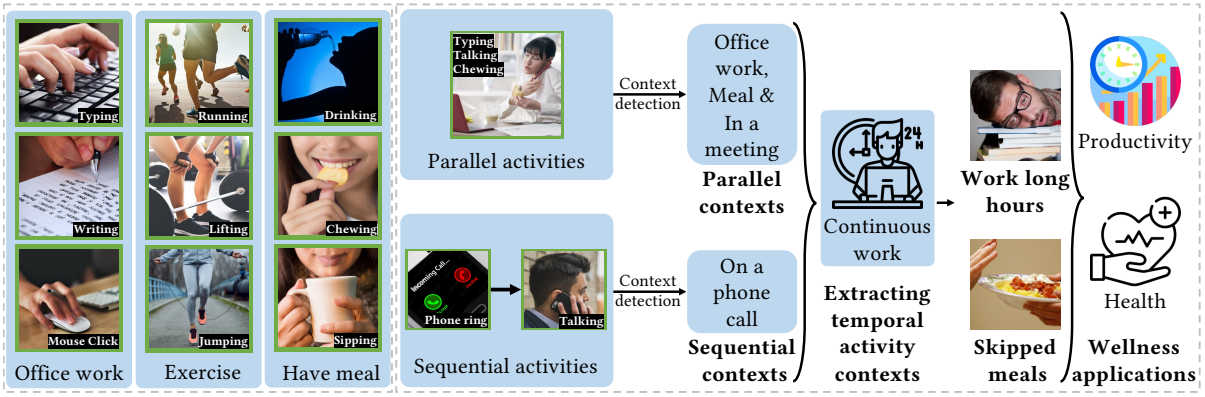


Fig. 1. Daily activities (e.g., *Activity A: typing, A: running, A: chewing*) can be translated into semantically richer contexts (e.g., *Context C: Office work*, *C: Exercise*, *C: Having meal*). However, real-world activity patterns are complex, with activities happening sequentially, in parallel, and interleaved, requiring a new approach to capture contexts. The TAO system is a hybrid system that combines ontological and temporal components to convert these activity patterns into rich contexts accurately. These contexts can then enable compelling applications, such as measures to detect and improve user wellness.

HAR-based approaches primarily focus on using the sensor data from smart devices (such as phones, watches, or ambient sensors) to accurately infer human activities over timescales in the order of seconds (such as *Activity A: talking, A: typing, A: jumping,* etc.), rather than extracting more semantically meaningful contexts (such as *Context C: Office work* or *C: Exercising*). Thus, understanding higher-level and semantically meaningful contexts of daily activities are crucial to support applications such as tracking productivity or wellness.

In recent years, a growing body of research around Context Aware (CA) computing has used ontology-based frameworks to model and reason about the context of human activities [2, 3, 15, 47, 48, 70]. These frameworks use Web Ontology Language (OWL DL, OWL 2 [25]) to define custom vocabularies (e.g., activities, location, or time) that are required to infer context and to model the logical associations between them. While these approaches provide a model of the complex relations of activities to infer context, the definition of the activity patterns is often static and highly structured. They are not flexible enough to represent real-world activities and their intricate temporal associations, leading to inaccurate context predictions. For instance, Figure 1 shows that when an individual is either *A: typing, A: writing, or A: clicking the mouse*, the ontology-based approaches infer the context as *C: Office work*. However, real-world activity patterns are complex. Activities such as *A: sitting* and *A: talking* may happen simultaneously or over a period of time that may relate to multiple contexts, such as *C: Office work*, *C: In a meeting*, and *C: On a phone call*. Similarly, an activity such as *A: walking* may be related to contexts such as *C: Commuting* or *C: Exercising*.

More recent efforts include hybrid approaches that combine knowledge- and data-driven techniques that derive semantic relationships between activities to detect context [28, 38, 51, 56, 58, 76]. In these approaches, statistical or clustering-based models are used to predict the most probable activities based on the sensor data, and an ontology is used to model the context based on the predicted activities. However, in these approaches, the accuracy of context recognition can be vulnerable to uncertainty and ambiguity due to inference from an incomplete ontology. Such approaches do not differentiate contexts from sequential (activity A happening after activity B) and parallel activities (activity A and B happening simultaneously) which predominantly occur in the real world. Moreover, most of these approaches focus on modeling specific contexts and as a result, do not support downstream context-based applications such as those for wellness.

We propose TAO, a hybrid system that leverages both an OWL-based ontology [25] and a temporal autoencoder-based clustering algorithm to detect semantically meaningful and richer contexts from complex real-world activity patterns (e.g., sequential and parallel). We design and build our custom ontology based on prior work [47, 70] and extend it to model various real-world activity patterns as complex activity relationships that can infer a wide range of contexts. For extensibility, our ontology uses SPARQL queries [61], allowing us to easily represent complex activity relationships and infer them to context definitions. We build a custom unsupervised clustering algorithm to model temporal contexts to identify recurring activity patterns and label these patterns using our ontology. TAO's hybrid approach handles sequential and parallel activities, accurately converting them into a rich set of contexts. Based on our evaluations, we are confident that the TAO system presents a versatile approach that can be applied effectively to a diverse range of human contexts with minimal modifications. Overall, we make the following contributions:

- We present the TAO system[1], a hybrid approach that combines OWL-based ontological techniques with temporal clustering methods to interpret semantically meaningful high-level contexts from low-level human activity patterns that happen sequentially, in parallel, or are interleaved.
- We implement an end-to-end TAO pipeline that takes in a set of human activities and provides contexts. We evaluate TAO on two large-scale real-world activity recognition datasets, *ExtraSensory* [67], and *Casas* [21]. We show that TAO achieves 80% and 87% accuracy, respectively (measured using the *Jaccard Similarity Coefficient*(JC)) for context detection of new activity patterns across multiple users, as compared to an accuracy of 11.03% to 20% for the state of art baseline approaches.
- We deploy TAO in the real world with 8 participants performing 17 activities in 4 different scenarios. We show that TAO can detect contexts from activities with 69%-76% accuracy (JC) in this setting.
- Finally, we prototype a wellness application that uses the rich contextual information from the TAO system to provide metrics that denote the stress and productivity of office occupants. We evaluate this application on *ExtraSensory* and *Casas* activity recognition datasets and show that the wellness metrics calculated by contexts provided by TAO are much closer to the ground truth (on average within 1.1%) as compared to a baseline approach (on average within 30%).

2 BACKGROUND AND MOTIVATION

Our overarching goal for TAO is to accurately translate fine-grained human activity patterns obtained from different activity recognition systems to meaningful contextual information that allows us to understand the user's behavior better. This contextual information can support various downstream applications in domains such as health, wellness, and human-computer interaction, including those that aid individuals in improving their quality of life, particularly those with cognitive or physical impairments. We motivate several potential use cases to illustrate the benefits of an accurate context-detection system.

2.1 Healthcare-based Applications

Context recognition in the healthcare domain can be used to monitor and support individuals with chronic conditions. For example, a system based on context recognition can help interpret a person's activities of daily living by looking at specific sequences of activities, such as *A: opening a medication box* followed by *A: drinking water* to detect adherence to medication. Similarly, inferring that an Alzheimer's patient left the kitchen to answer an incoming call (*A: phone ring*) but failed to resume lunch (*A: eating*) while in the context (*C: Eating a Meal*) afterward can help with memory augmentation tools. Such contextual information could provide reminders or prompts to help the person maintain a healthy routine and alert caregivers if they are not engaging in activities necessary for their well-being while reducing the number of false alarms. Similarly, wearable trackers and medical

[1] www.github.com/synergylabs/tao

devices monitor a person's current activity and vital signs, such as heart rate and blood pressure. These signs can vary depending on the patient's activity level and other factors. With context recognition, a system can understand the context in which the vital signs were measured (e.g., *C: Exercising* or *C: Office work*), allowing for more accurate attribution and interventions. For example, a change in the vital signs in one context may be acceptable while being a sign of stress in another. Understanding such contexts would be very useful in providing more targeted and actionable interventions for addressing mental health issues [17].

2.2 Smart Building Applications

In smart building scenarios, a potential application of context recognition is to improve productivity and reduce stress. For example, suggesting tasks or actions based on the person's current context, providing reminders and prompts to help them stay on track, and automatically adjusting the environment to create a more conducive work environment. For example, if the system detects that an office occupant is in the same context for long hours, such as *C: Office work*, while engaging in multiple activities, such as *A: talking* or *A: typing*, it can nudge the user to take a break to improve their productivity. Moreover, such a system in office buildings could improve energy efficiency and comfort for building occupants by changing environmental parameters based on their context. For example, if the context is identified as *C: In a meeting*, the HVAC system of the room can be configured to allow more airflow into the office. Similarly, when the occupant is in the context of *C: Working* for some time, denoting focused work, their status can be set to busy automatically on software such as Slack to prevent interruptions. In contrast, when the context detected is *C: Taking a break*, they can be marked as available to allow for impromptu social interactions. Note that a user in each of the example contexts above could be doing a wide variety of low-level interleaved activities (e.g., A: talking, A: using the PC, A: pacing, A: typing, etc.).

2.3 Smart Home Applications

In a smart home setting, detecting context can be powerful to enable new scenarios that are impossible based on detecting activities alone. For example, consider the context of *C: Preparing a fresh meal*, which can be represented by different activities such as *A: cutting, A: chopping, and A: washing vegetables*. In contrast, the context of *C: Preparing a pre-cooked Meal* can be represented by activities such as *A: using Microwave, A: opening the refrigerator*. A home user can use these detected contexts to get an idea of how healthy their diet is and get nudged by a potential app to buy fresh groceries. In addition, such a system could use sequential activity patterns, such as *A: cutting, A: chopping, A: washing vegetables*, etc., to track the progress of the user through a recipe, providing step-by-step instructions and alerts for important steps, such as when to add ingredients or when to check the temperature of the oven. Such a system can make cooking easier and more efficient, helping users avoid mistakes and stay on track. As another example, detecting the context of *C: Sleeping* could automatically enable the home alarm and disarm it when the home occupant wakes up. As another example, an activity detection system could detect simultaneous activities such as *A: talking* and *A: eating* in the living room, which may denote multiple potential contexts *C: Watching TV*, *C: Amusement*, or even *C: Relaxing*. Depending on the context, a smart home control app could suggest dimming the lights for an optimal viewing experience or keep the lights on.

3 RELATED WORK

Several prior works have proposed different modeling approaches for context reasoning architectures primarily focusing on activity recognition. Existing research in this space falls into three broad categories: (a) knowledge-driven approaches that focus solely on ontological reasoning; (b) data-driven approaches that rely on modeling temporal activity patterns, and; (c) approaches that combine knowledge and data-driven approaches. We refer the reader to [7, 53, 54] for an extensive survey on this topic and compare several prior works with our TAO system.

Ontology-driven Approaches: Researchers have proposed numerous ontology-driven approaches to model the semantics of low-level activity information and recognize user context [2, 3, 15, 16, 47, 48, 70]. Most ontological approaches use knowledge representation languages such as OWL [25] to define generic vocabularies for individual domains using low-level activity definitions. In such approaches, the context is represented as a set of axioms about entities and resources that are further associated through properties and relationships, providing a uniform representation of data. For instance, approaches in a smart home [15, 16, 70], contexts correspond to OWL individuals, and realization is used to determine into which context concepts a specific situation individual falls into. In addition to using an ontology, other approaches such as Context Aggregation and REasoning (CARE) middleware also use statistical reasoning for context inference (e.g., business meeting) based on the location or environment (e.g., office) and with at least two actors (e.g., employees) [2]. Other approaches extend their ontology such that their OWL 2 reasoning module incorporates temporal correlations of complex activities using rules and well-defined SPARQL queries that are essential in context recognition [47, 48]. While this approach to modeling complex relations between activities as context is useful, the definition of numerous activity patterns is often static and highly structured. In contrast, the TAO system allows us to model and reason over intricate, simple temporal dependencies between activities that indicate activity patterns that are sequential and parallel.
Temporal Clustering Approaches: Prior research has also proposed data-driven methods, such as using temporal clustering algorithms to identify patterns in multivariate time series data. These approaches focus on identifying sensor data patterns from multi-modal sensors to detect activities in space [35, 37, 41, 69]. Other approaches, on the other hand, focus on directly modeling individual behavior patterns (such as movement and activity routines) using sensor data from mobile and/or ambient sensors [1, 32, 34, 44, 55, 81]. Other temporal clustering methods aim to identify a complex relationship between simple activity sequences when demonstrating a procedure such as *"how to change a tire"*[23, 24, 39, 75]. Their primary focus is to extract procedural knowledge about a particular kind of long-term activity (i.e., cooking, building models, etc.) to enable skills for AI agents[19, 82] or to understand human psychology[26, 45]. Other approaches use probabilistic and statistical methods to model temporal activity patterns and identify abnormal human behaviors [5, 42, 59]. While these data-driven pipelines identify activity patterns, they are limited to applications such as identifying anomalies and do not focus on capturing the semantic context of these activity patterns. Recent approaches have proposed ML-based methods that use fine-grained sensor data to measure wellness indicators such as mood instability [49], productivity, and stress [4, 36] without requiring the need for context detection. However, such approaches rely on fine-grained data from several input sources, such as cameras, wearables, and smartphones. As a result, these approaches require manual user input to self-report productivity or stress every hour, which can be cumbersome. In comparison, TAO proposes an ontology-based approach to predict contexts accurately and automatically, which can then be used to calculate wellness metrics.
Hybrid Approaches: Several recent efforts aim to combine both knowledge-driven and data-driven approaches to derive semantic relationships between activities for context inference [28, 38, 51, 56, 58, 60, 76]. COSAR [56] uses a statistical classifier that recognizes an activity which is then used to obtain context information using the ontological reasoner from the ActivO ontology models [57]. In other approaches [28, 51, 76], the sensor data is segmented based on activity relationships inferred from an ontology, and a clustering model is trained to capture temporal patterns related to a context. These hybrid approaches are closely related to our TAO system, showing the promise of data-driven pre-processing for activity inference in combination with ontology to improve context recognition. Such approaches, however, are easily affected by the noisy nature of sensor data streams and events, resulting in inaccurate context recognition. A more recent approach utilizes a combination of an Ontology, Computational Causal Behavior Models (CCBMs), and Hidden Markov Models (HMMS) to identify goals for cooking activity, including the type of cooking and healthy vs. unhealthy meals [77]. Extending this approach across diverse contexts becomes challenging, as the models are constrained to specific sensing modalities and contexts (e.g., limited to activities within the kitchen environment). Further, their wellness metrics are limited to

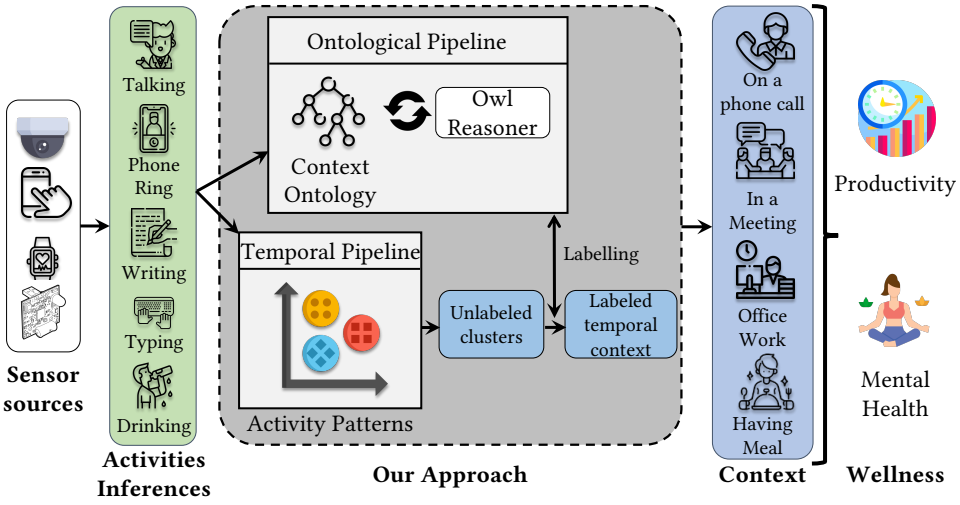


Fig. 2. Overview of the TAO's system architecture. TAO leverages OWL-based ontologies and temporal clustering approaches to identify context from the stream of activities obtained from Human Activity Recognition (HAR) systems. The contexts detected by TAO are then sent to our wellness application which then infers productivity and stress.

healthy vs. unhealthy eating habits, whereas TAO can generalize to wellness metrics like productivity and stress that relies on a richer set of contexts.

Gap in the Existing Systems: In summary, none of the prior work considers complex and wide-ranging daily life activity patterns that are often sequential or parallel in nature in order to detect semantically meaningful and rich contextual information accurately. In addition, as far as we are aware, none of the prior works are agnostic to the underlying human activity recognition (HAR) models and only depend on activity predictions over time to extract context information. Finally, none of the prior works have used contexts extracted from daily life activities to build end applications that can indicate metrics such as productivity or stress.

4 TAO: SYSTEM ARCHITECTURE

Figure 2 shows the system architecture of TAO highlighting two key components, namely the ontology pipeline (§4.1) and the temporal pipeline (§4.2). Our ontological pipeline provides a standard vocabulary for modeling activity-related information, such as domain activity classes, actors, etc., and for formally describing the complex relationships between the activities (activity patterns) as Contexts (§4.1.1). In addition, it also provides a method to derive contexts from complex activity patterns using the OWL 2 reasoning paradigm and the execution of SPARQL CONSTRUCT queries (§4.1.2). Our temporal pipeline (§4.2) is responsible for learning context representations based on the complex activity patterns that happen over a period of time (e.g., 5mins, 10mins, 30mins, 1hr). To enable this, our temporal pipeline uses a novel featurization technique to model activity relations and a deep-learning-based unsupervised learning approach to model activity patterns. The unlabelled clusters in our temporal pipeline utilize the ontological pipeline to label these representations to infer existing and new contexts. Overall, activity inferences from any human activity recognition pipelines can be fed into TAO, specifically the ontological and the temporal pipelines, to assess the activity patterns and generate semantically meaningful and richer contexts that can be used for applications such as indicating wellness such as productivity or stress. We describe each of these components in further detail.

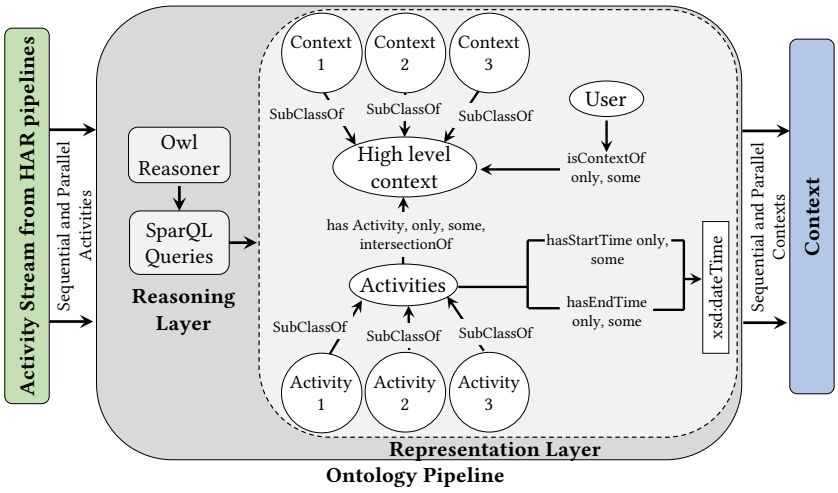


Fig. 3. Overview of TAO's Ontological Pipeline.

4.1 Ontological Pipeline

The architecture of our ontological pipeline is shown in Figure 3. It comprises a Representation Layer and a Reasoning Layer. We designed our representation layer such that it consists of a dense representation of complex activity patterns which we see in daily life and can be interpreted into contexts. In addition, we wanted to ensure that we modeled the context information that is accessible using simple queries of activity relationships. This is primarily useful in cases where we want to interpret a context based on a complex activity relationship that consists of several constraints, such as time duration or order of important activities. To enable this, we built our custom ontology based on the prior work[47, 70] to provide a dense vocabulary for formally describing the context as complex relations among activities. We describe the representation layer further in §4.1.1. We design our reasoning layer such that we can derive contexts by interpreting the relationships between the defined activities in our custom ontology. The reasoning layer uses an OWL 2 reasoner [25] to model complex activity patterns, e.g., class subsumption, sub-properties, inverse properties, etc., while the context inference is realized by custom-defined SPARQL queries [61]. We describe the reasoning layer further in §4.1.1.

4.1.1 Representation Layer. The goal for the representation layer is to design an ontology to provide a standard definition of vocabularies that model contexts as relationships between complex activity patterns that happen in our daily life. These contexts include (1) contexts based on sequential activity patterns, (2) contexts based on parallel activity patterns, and (3) contexts based on a variation in the duration of activity. Figure 1 shows an example of a context *C: On a phone call* which requires the *A: phone ring* activity to be sequentially followed by the *A: talking* activity. Similarly, often multiple activities may occur simultaneously, such as *A: typing*, *A: talking*, or *A: chewing*, which indicates that multiple contexts are happening at the same time, such as *C: Office work*, *C: Having meal*, and *C: In a meeting*. Moreover, context inference also varies based on the duration of activities. For example, both contexts *C: On a Zoom call* and *C: On a phone call* require the activity patterns to be sequential (*A: phone ring* followed by *A: talking*); however, the duration of the activity *A: talking* (5mins vs. 30mins) differentiates these contexts. Thus, to model these complex activity patterns as contexts, we designed the representation layer with a custom ontology that incorporates the concepts proposed in the Meeting Minds Ontology [70] and the lightweight Domain Activity Ontology [47]. Specifically, we leverage the vocabularies defined by the Meeting Minds behavioral scientists [70] for contexts as activity combinations and the definition concepts from Domain Activity Ontology [47] to enable capturing of time duration information of activities. For example, while Meeting

```
CONSTRUCT {
?x a :OfficeWork .
    :hasActivity ?x;
    :hasActivity ?y;
}
WHERE{

?x a typing;
?y a talking;
BIND((URI(?x, ?y) as ?new) .
NOT EXISTS (?new a [] .)

}
```

Fig. 4. SPARQL query to infer *A: typing*

```
CONSTRUCT {
?x a :OnAPhoneCall .
}
WHERE{
    ?x a phoneRing;
        :hasStartTime ?st;
        :hasEndTime ?et.
    ?t a talking;
        :hasStartTime ?st1;
        :hasEndTime ?et1.
    FILTER (contains(?st, ?et))
    FILTER (before(?st1, ?et1))
}
```

Fig. 5. SPARQL query to infer *C: On a phone call* context

Minds only provides a context *C: House work* as a relationship between two *A: standing* or *A: walking* and *A: sweeping* occurs, the DAO concepts allow us to define activity time duration, such as T(A: standing)$_{end}$ → T(A: sweep)$_{start}$. Using this combined approach and including the activities obtained from the activity recognition datasets (*Extrasensory and Casas*) which were not part of the combined ontology, we expanded our ontology from 16 activities in [70] to 88 activities. To ensure the ontology accurately reflected the complexities of real-world scenarios, we meticulously designed and modeled approximately 900 rule definitions. To bootstrap the context labeling process for the ontology, the researchers were shown patterns of activity combinations timeline view (similar to 18), and multiple researchers jointly and independently labeled the activity to the context mappings. This process allowed us to ensure validity in the context of mappings, which is similar to prior approaches [47, 70]. These rules were formulated to capture the relationships between activities and encompass various context scenarios involving activities occurring in sequence or in parallel. Furthermore, we included rules that accounted for time-based constraints, allowing us to consider temporal aspects when inferring context. Overall, using these combined approaches allowed us to model a wider range of context as an intersection of multiple activities and relationships beyond what was possible by these individual approaches. Figure 3 shows the overview of the ontology where the activities of daily life are represented as instances of the *Activity* class, and they are linked to ranges of time through the use of the *hasStartTime* and *hasEndTime* datatype properties. The instances of the *Contexts* class model the relationship between the instances of the *Activity* class using the object properties *hasActivity* along with *intersectionOf, some, only, union* object definitions. For example, the context *C: Office work* is an instance of the class *Context* and models the relationship between instances of the *Activity* class such as *A: typing* and *A: writing* using *intersectionOf* object properties.

4.1.2 Reasoning Layer. Our reasoning layer derives context by meaningfully interpreting the relationship between the primitive activities specific to the context using a combination of standard OWL reasoners and SPARQL queries. The OWL reasoners determine whether or not the ontology is consistent and identify subsumption relationships between classes, such as Context or Activities. The SPARQL query language then allows customized queries to interpret Context. We used this combined approach to design our reasoning layer primarily because an OWL reasoner by itself cannot support query-like operations that are required to interpret contexts. Combining both the reasoner and SPARQL queries, on the other hand, can render the context of an activity or multiple activities easily. Thus, TAO uses the OWL 2 reasoner to formulate the context and activity relationships, and we then use the SPARQL queries to query the combined pieces of information (intersections). Specifically, the ontology semantics, e.g., property restrictions, sub-properties, inverse properties, etc., are handled by the OWL 2 reasoner, whereas domain-specific SPARQL queries realize the context recognition.

4.1.3 SPARQL Context Interpretation Queries. The SPARQL query language enables ease for querying graph patterns along with their conjunctions and disjunctions. The SPARQL in the TAO system is defined in terms of a CONSTRUCT and a WHERE clause: the CONSTRUCT clause defines the graph patterns, i.e., the set of information that should be returned to upon the successful pattern matching of the graph in the WHERE clause. We show two examples of graphs-based SPARQL queries in Figures 4 and 5. Figure 4 shows how we query handles the composition semantics of *C: Office work* context, using the classes *A: typing* and *A: talking* as its sub-activities. Similarly, Figure 5 shows modeling of *C: On a phone call* context using the *A: phone ring* and *A: talking* as sub-activities with time duration information of how long each activity would be performed. These queries can be updated to infer context based on complex dependencies of underlying activity relationships.

4.2 Temporal Clustering Pipeline

Ontological approaches work well to detect instantaneous contexts based on short sequential patterns or parallel occurrences of activities. However, in real life, we regularly switch between contexts. At one time, we can be in multiple contexts, i.e., reading and having a meal at the same time or listening to music while vacuuming, etc. These settings with multiple contexts provide an opportunity to derive richer context information by capturing repetition across interleaved sequential and parallel activity patterns. We present a temporal clustering-based approach to detect such interleaved and recurring patterns. We define activity detection pipeline as a set of (one or more) models that utilized raw data from any sensor sources (see Figure 2) and outputs instantaneous activity inference(s) for a given timestamp (i.e., typing, talking, drinking, etc.). These outputs are combined into a single activity stream which consists of *timestamp-activity* inference pairs. This activity stream is an input for our context prediction model. Further, we parameterize the input activity stream with two parameters, (a) *stacking window*, and (b) *lag window* (see Figure 6). We define the *stacking window* as the time interval to wait for receiving activity inferences. Typical time intervals for a *stacking window* are from a few seconds (1 second, 5 seconds, etc.) to a few minutes (1 minute, 5 minutes, etc.) based on the output rate of the activity detection pipeline. Activities received in one *stacking window* are considered to happen in parallel . i.e., in a sample activity stream shown in Figure 6, we see that we receive one activity inference between 0-2 minutes (*A: eating*), no activity inferences between 4-6 minutes, and multiple activity inferences between 8-10 minutes (*A: drinking* and *A: writing*). To capture patterns over the sequence of activities, we observe the incoming activities for a time interval (defined as the *lag window*), which is longer than the *stacking window*. Typically, a *lag parameter* is set to a multiplier (5x, 10x, 30x, etc.) of the *stacking window*.

The sequence of (a set of parallel) activities happening in a *lag window* is used for context prediction at the end of the *lag window*. Figure 6 shows how we encode activities in a *stacking window*(t) as a binary encoding A_t of size N, where N is a set of all possible activities, with each positional argument being 1/0 based on whether that activity is detected. Next, we stack these binary vectors horizontally for a *lag window* (T) to create a sparse representation of activity patterns X_T seen in the *lag window*. This is the final input to our temporal clustering (See Figure 8), which consists of three components, (i) *Representation Trainer* which learns meaningful embeddings from context representation X_T, (ii) *Cluster Trainer* which clusters these embeddings into distinct context clusters, and (iii) *Context Labeler* which derives labels for prominent context clusters using the ontology reasoning layer.

4.2.1 Representation Trainer. A naive way to identify recurring temporal patterns is to cluster these sparse representations directly. However, positional dependence of activities leads to significantly different representations for similar contexts. i.e., *A: jumping* followed by *A: jogging* would be quite different from *A: jogging* followed by *A: jumping*. One way to reduce this bias is to featurize context representations to encode sequential and parallel patterns explicitly. Finding a fixed approach for such featurization is not generalizable as the importance of such highly localized patterns can differ for different user sets. Rather, we built a *Representation Trainer*, which learns to featurize our input context representation X_T (see Figure 6). It utilizes an autoencoder-based approach, an

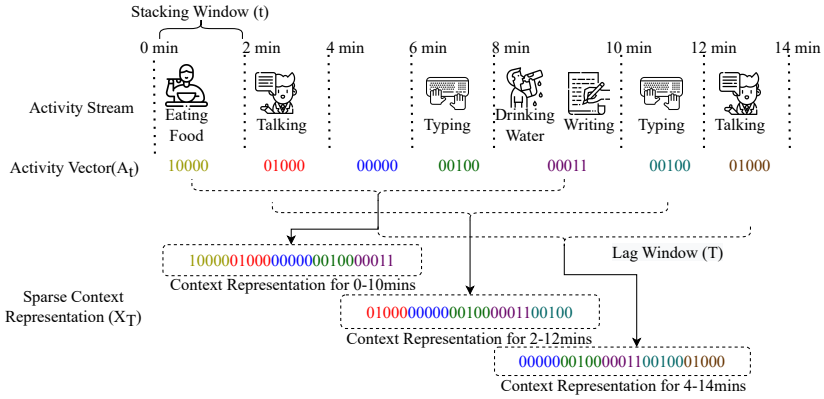


Fig. 6. An example of context representation (five activities only) for 2 min stacking window(t) and 10 min lag window(T).

unsupervised neural network that learns how to compress and encode data efficiently and then how to reconstruct the data from the reduced encoded representation to a representation as close to the original input as possible [79]. The encoded data, i.e., dense context embedding Z_T, is further used as initial input for our *Cluster Trainer*. A basic autoencoder [71] maps input to a denser latent representation but does not guarantee to encode sequential or parallel patterns. Another baseline approach is to use Long Short Term Memory (LSTM) for modeling sequential patterns [30]. However, based on our initial experiments when designing the temporal pipeline, we found that LSTM-based autoencoders failed to learn context representation effectively for our use case. One major reason is that LSTMs need dense input data and longer sequences to learn meaningful representations. Thus, they fail to model temporal patterns effectively from sparse context representation.

Based on our findings from baseline approaches, we used a specialized network architecture, Temporal AutoEncoder (TAE) [46], that creates dense embeddings, encoding temporal information over sequences of activity stream in X_T. TAE can effectively capture temporal relationships in multivariate time series data. Figure 7 shows an architecture for the TAE encoder and decoder. The first level of network architecture consists of a 1D convolutional layer, which extracts short-term features (waveforms), followed by a max pooling layer. This casts time series into a more compact representation while retaining the most relevant information. This step helps learn local patterns in the sparse context representation, thus modeling information from activities happening in parallel. These compact representations are then fed into the second layer, consisting of two bi-LSTM cells, to learn temporal changes in both time directions. The Bi-LSTM layers help in capturing sequential patterns of activities in final embedding. Finally, reconstruction is done by an upsampling layer followed by deconvolution later to obtain a reconstructed signal. This architecture has been tested with time series information across various domains in literature [46]. Our *Representation Trainer* pre-trains a TAE architecture with a binary cross entropy [22] loss function to learn context embeddings Z_t (More details in Section 5). These context embeddings are provided as an initial representation to derive optimal cluster count and later cluster a variety of activity patterns into context clusters with our *Cluster Trainer*.

4.2.2 Cluster Trainer. A direct approach to derive contexts from embeddings Z_T is to use partition or hierarchical clustering approaches to derive prominent contexts based on cluster centers. However, these approaches do not utilize data-specific patterns in activity streams (X_T) to enhance separability in centroids for better clustering. Existing literature in clustering methods has shown superior performance by jointly training autoencoder for latent representation and clustering error[80]. Our *Cluster Trainer* uses a similar, combined training approach for cluster assignment, which is inspired by Deep Temporal Clustering (DTC) [46] and Deep Embedded Clustering

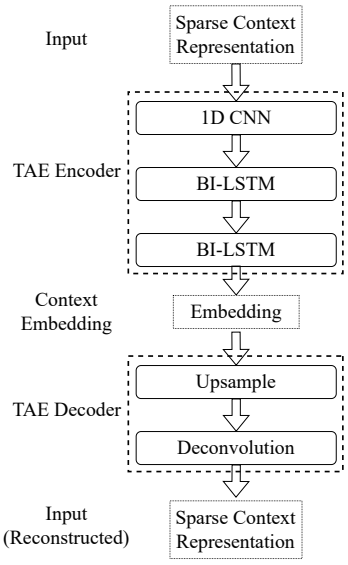


Fig. 7. Architecture of temporal autoencoder (TAE).

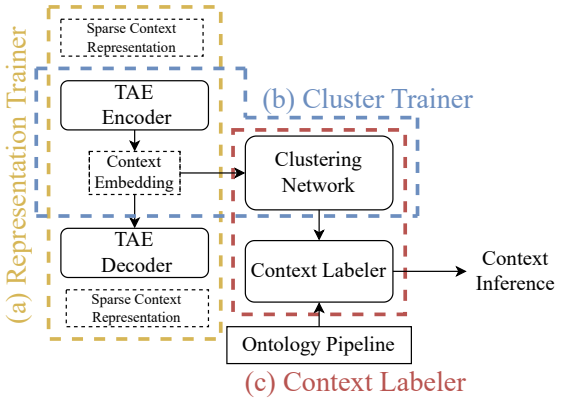


Fig. 8. Overview of TAO's Temporal Pipeline. Representation Trainer (a) pretrains TAE Encoder-Decoder to learn dense context embedding from sparse input representation, then Cluster Trainer (b) uses TAE-Encoder and learned embeddings to discover optimum context cluster count and their sparse representation, and finally Context Labeler (c) uses ontology to provide labels to generated context clusters.

(DEC) methods [74]. In this approach, we iteratively update cluster centroids to optimize for the separability of clusters and fine tunes weights in the *Representation Trainer* to ensure that learned context embeddings are best suited to separate context representations X_T into given categories.

One of the challenges of using such a combined training method is to figure out an optimum number of clusters. As the number of recurring activity patterns is different for different settings, we need a method to figure out cluster count based on given context representations X_T. Spectral clustering methods (i.e. DBSCAN, OPTICS, HDBSCAN, etc.) provide a natural way to figure out optimal cluster count based on other parameters like distance threshold for neighboring points, minimum clusters in a sample, etc. However, we observed that spectral clustering methods are sensitive to the subspace of embeddings Z_T and their hyperparameters. For different kinds of activity streams and length of context representation, underlying subspace Z_T varies significantly, thus leading to a wide range for the number of optimal clusters, whereas the number of meaningful interleaved patterns for a given set of activity streams is finite. Thus, we used a simpler method using silhouette scores[78] with K-Means clustering over context embeddings Z_T. One caveat of using this method is that partition-based clustering assumes convex cluster boundaries, which is alleviated by fine-tuning cluster membership of context representations X_T based on the deep clustering method in the next step.

Next, we initialize centroids with K-Means clustering with optimal cluster count over learned context embeddings Z_T, followed by iterative training with an unsupervised method that alternates between two steps till it ends. First, we compute q_{ij}, which is the probability of assignment of input X_i belonging to cluster j based on the distance of context embedding Z_i and centroid w_j. The closeness is evaluated using complexity invariant distance (CID) [6], thus allowing a generalizable distance metric across various complex activity streams. It is based on a Euclidean distance (ED), corrected by the complexity factor (CF) of two representations (Z_i and w_j)

$$dist(Z_i, w_j) = CF(Z_i, w_j) * ED(Z_i, w_j)$$

$$CF(Z_i, w_j) = \frac{max(CE(Z_i), CE(w_j)}{min(CE(Z_i), CE(w_j))}$$

$$CE(x) = \sqrt{\sum_{t=1}^{N-1}(x_{t+1} - x_t)^2}$$

where $CE(x)$ is the complexity estimate of a sequence x. We normalize these distances $dist(Z_i, w_j)$ into probability assignments q_{ij} using a Student's t distribution kernel with a single degree of freedom [68].

$$q_{ij} = \frac{(1 + dist(Z_i, w_j))^{-1}}{\sum_{j=1}^{k}(1 + dist(Z_i, w_j))^{-1}}$$

Second, we train the *Cluster Trainer* iteratively by minimizing KL divergence loss between q_{ij} and target distribution p_{ij} to strengthen high confidence predictions and normalize the losses to prevent distortion of context embeddings, which is

$$p_{ij} = \frac{q_{ij}^2/f_j}{\sum_{j=i}^{k} q_{ij}^2/f_j}$$

where $f_j = \sum_{i=1}^{n} q_{ij}$, which is derived empirically [29, 74]. Finally, we compute KL divergence loss:

$$L = \sum_{i=1}^{n}\sum_{j=1}^{k} p_{ij} log \frac{p_{ij}}{q_{ij}}$$

We use a weighted combination of KL divergence loss and *Representation Trainer* loss to optimize centroids and fine-tune the embedding encoder weights. This provides us the final cluster centroids $w_j^{*(z)}$ in the subspace of context embedding Z_T, which are fed back into the *Representation Trainer* to reconstruct $w_j^{*(x)}$, a representation of cluster centroids in the subspace of sparse context representations(X_T's).

4.2.3 Context Labeler. Finally, $w_j^{*(x)}$ is the context representation for each cluster and defines the context for all X_T belonging to cluster j. However, when converted into timestamped activities A_t^*, these representations can be unnecessarily long and interleaved with activities from multiple contexts. A simple example can be a cluster representation shown in Figure 6 between 4-14 minutes, i.e., *A: typing* > *A: drinking+A: writing* > *A: typing* > *A: talking*. We use a heuristic method to break this sequence of activities and run multiple queries through our ontology to detect semantically meaningful contexts.

A naive approach is to query our ontology to provide context for all possible subsets of sequential, parallel, and single activities. It will provide multiple contexts (i.e., for our example *C: In a meeting*, *C: Having meal*, *C: Office work*, etc., whereas the actual context is only *C: Office work*). To reduce inaccurate contexts, we use two heuristics that work well in practice. First, we only find contexts from all sequential and then parallel activity patterns of two activities. i.e., in our example, an ontology query for (a) *A: typing*, (b) *A: drinking+A: writing*, and (c) *A: typing* > *A: talking* will return *C: Office work*. If we do not find any meaningful contexts, we find contexts based on single activities in the set of interleaved activities. Second, we use our ontology to find the subset of activities that identify to a particular context (i.e., only *A: typing* would indicate the *C: Office work* context, in contrast to only *A: talking* could indicate multiple contexts like *C: Having meal*, *C: Office work*, etc.). We attempt to find context labels based on interleaved sequences of these subsets of activities first. If we do not find meaningful contexts, we combine the contexts from all possible subsets of sequential, parallel, and single activities.

If new activities are found in activity streams not modeled in the ontology, the ontology cannot provide contextual information even with the entire subset of relations. In such cases, we can ask users to label the context based on their understanding manually. These patterns and provided contexts can be used to update the

ontology to capture a richer set of contexts over time. We keep this as a future extension of our work and scope our current work to a set of activities modeled by our extended ontology.

4.3 Putting it All Together: The TAO System

We combine context predictions from the ontology and temporal pipeline to give final context predictions. A simplistic way to combine contexts from both pipelines is to take the union of unique contexts predicted for a given time period. However, this method leads to an overestimation of the context for activities. One primary reason for overestimation is the presence of ambiguous activities(i.e., activities that span across multiple contexts like *A: Sitting* or *A: talking*), which leads ontology to predict all the possible contexts, i.e., *C: Office work*, *C: Amusement* and *C: Having meal*, etc. To alleviate this issue, we opportunistically select output from one of the pipelines based on a simple metric, the number of contexts predicted by each pipeline for a given time period. An ontology is deterministic in nature as it consists of pre-defined rules for mapping activities to contexts. Thus, the ontology pipeline provides more reliable context detection than the temporal pipeline for scenarios where we only see a single activity in the activity stream or a well-defined sequential set of activities. On the other hand, the temporal pipeline is more reliable in a multi-context setting, as it predicts multiple contexts only when corresponding activity patterns are recurring in activity streams. In comparison, the ontology provides multiple context predictions due to the presence of ambiguous activities (like *A: Sitting* or *A: talking*). Thus, we choose an output from the temporal pipeline when there are multiple context predictions from either of the pipelines.

5 IMPLEMENTATION

Our TAO system [64] is implemented to take activities as inputs from any activity recognition pipeline. TAO then takes these activity predictions and sends them to both our ontological and temporal pipelines, returning the detected context(s). We developed our TAO ontology using OWL 2 RL reasoning [25] using SPARQL queries (SPARQL Inferencing Notation — SPIN [72]) for implementing the context interpretation process. We develop, build and edit our ontology using in Protégé [50] using the Pellet reasoner and Hermit reasoner. Our temporal pipeline is written in Python, and our deep learning models are built using Pytorch[52]. We utilized rapids framework [65] to support faster K-Means clustering on a GPU for finding optimal cluster count and initializing centroids for the *Cluster Trainer*. Our system uses five different submodules, (i) *Data Parsers*, which parse different types of data streams into a common input format (i.e., sparse context representation) for training, (ii) *Representation Trainer*, which learns the context embedding and locally stores pre-trained autoencoder models, (ii) *Cluster Trainer*, which embeds pre-trained autoencoder models into a temporal clustering network to extract context embeddings, as well as learn optimal cluster counts and cluster assignments, (iv) *Context Labeler*, which stores a copy of the most updated ontology and utilizes all of the above components to label clusters to a specific context(s) and (v) *Prediction Engine*, which is used to deploy this trained pipeline for on the go context predictions. For training, TAO takes a JSON configuration file as an input, containing (i) data specifications and (ii) model specifications. Data specifications include the list of activities and contexts, method of streaming (i.e., POST requests or reading from a static file), *lag parameter*, *stacking window*, and ontology models. Model specifications include learning rate, context embedding size, TAE model parameters (i.e., layer sizes) for *Representation Trainer*, min/max bounds for optimal cluster count, learning rate, and stopping criteria for *Cluster Trainer*. Based on our experiments, we provide default model specifications to enable out-of-the-box training for any activity streams of type <*timestamp, list of activities*>. The final output for our training pipeline is a zip package, which can be deployed as a prediction service to serve context predictions given a stream of activities. We open-sourced our entire TAO system [64], including our ontology models, which can be used out of the box with activity streams from various kinds of sensing systems for context prediction.

6 EVALUATION

In this section, we evaluate the TAO's ability to accurately identify context from different activity patterns using representative HAR datasets as well as a real-world study. Our evaluation aims to answer the following questions:

- TAO's goal is to *accurately* detect contexts from different activity patterns. *RQ1: How well do the components of TAO, namely the ontological and temporal pipelines, detect contexts from daily activities? How well does TAO's hybrid approach that combines both pipelines detect contexts?*
- In addition to accurate context detection, TAO can also detect *richer* set of contexts more accurately than prior work. *RQ2: How well do we perform compared to prior work?*
- TAO is also *robust* in detecting contexts in real-world settings. *RQ3: How well does TAO perform in a real-world deployment, and how does the accuracy of an underlying activity detection pipeline impact TAO's performance?*

We describe our experimental setup and define the metrics we use to characterize accuracy in Section 6.1. In Section 6.2, we evaluate the accuracy of the individual components of TAO, namely, the ontological and the temporal pipeline and TAO's overall accuracy. We compare our system with prior approaches in Section 6.3. Finally, in Section 6.4, we evaluate TAO in a real-world study across 8 participants performing four different activity scenarios to show how our system detects a rich set of contexts with high accuracy.

6.1 Evaluation Setup

To evaluate our TAO system, we benchmark it on two human activity datasets (*Extrasensory* [67] and *Casas* [21]) and on a real-world activity detection pipeline. The dataset-based evaluation allows us to evaluate the performance of TAO on continuous activity streams across multiple days (longitudinal study) and compare TAO with other approaches that have also been evaluated on the same datasets. The *Extrasensory* dataset consists of activity and location labels of smartphone sensor data from 60 participants across several days (about 7-14 days per participant). The activity labels per participant are at 1-minute intervals, and there are 34 unique activity labels across all participants. The *Casas* dataset collects data at 1-second granularity from ambient sensors deployed in 30 different smart homes across 1.5 months. This data is manually labeled by researchers, and the entire dataset consists of 42 unique activity labels.

6.1.1 Ground-Truth Context Annotation. Most public datasets have activity annotations on various sensor data. A few public datasets have context annotations from sensor data. But, none of these datasets have an activity for context mapping. Thus, we created our own context labels using annotated activities from these datasets. We used the activity labels to create timestamped activity traces (at a participant level in *Extrasensory* and at a home level in *Casas*). We identified 14 different contexts from these activity labels, and two researchers independently annotated 90 days(randomly sampled) of activity data across both datasets. We annotated a total of 1300 hours of data across 1800 annotated instances of contexts. Detailed summaries about dataset characteristics and context annotation are added in the appendix (Section A).

6.1.2 Performance Metrics. To evaluate the accuracy of TAO, we use Jaccard similarity Coefficient (JC) [33] (see Figure 9). For context-level performance, we use precision and *recall*, which is calculated based on True Positives (TPs), False Positives (FPs), True Negatives (TNs), and False Negatives (FNs) at a context level. A specific detection for a context (say *C: OfficeWork*) is considered as TP if it is both detected and present in the ground truth, an FP if it is detected but not present in the ground truth, and an FN if it is not detected but present in the ground truth. The precision and recall metrics at the context level are then calculated as $P = TP/(TP + FP)$ and $R = TP/(TP + FN)$. For overall accuracy, we use a weighted average of the *Jaccard Similarity Coefficient* (JC) across all detected overlaps weighted by the length of the overlap interval. We also show the weighted average of

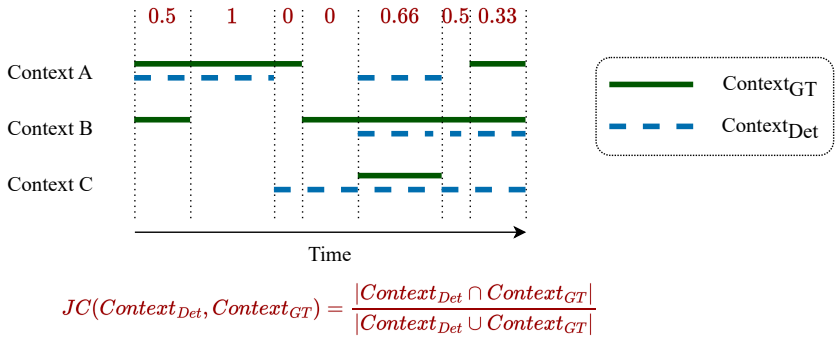


Fig. 9. A representative figure to understand the evaluation of context prediction accuracy from predicted contexts, $Context_{Det}$ when compared to ground truth, $Context_{GT}$, using Jaccard Coefficient (JC). Values on the top show Jaccard Coefficient for different overlap intervals. The JC ranges from 0-1, i.e., 0 when there is no overlap between ground truth and detected contexts and 1 if ground truth and detected context(s) are identical. Further, JC penalizes methods that detect contexts that are not present in the ground truth (larger value of union of sets) and for not detecting some contexts present in the ground truth (smaller value for the intersection of sets).

precision and recall at the context level, weighted by the number of times a particular context appears in the ground truth. In addition, we calculate the F1 score metric to compare our system with prior approaches.

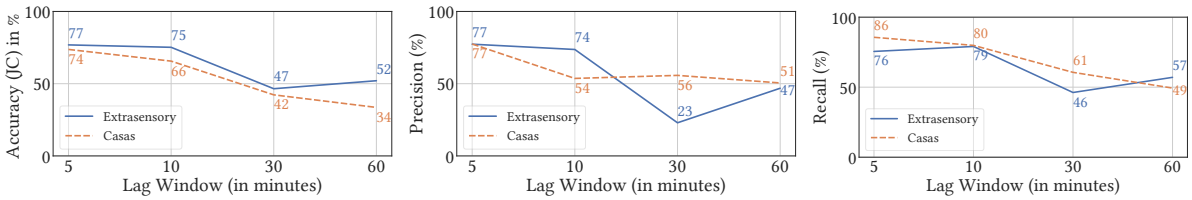
6.2 RQ1: Evaluation of TAO's Ontological, Temporal and Hybrid Pipelines

We evaluate the components of TAO, the ontology, and the temporal pipeline to measure their accuracy performance in isolation. For this evaluation we use the two HAR datasets and our ground truth labelled contexts. Specifically, we highlight the differences in terms of accuracy (JC), precision, and recall of context prediction, as applicable, when we use the TAO system with other datasets.

Table 1. Comparing the accuracy (JC) and the total contexts detected by TAO's ontological pipeline only with prior work on two real-world HAR datasets. Our approach detects different types of contexts (sequential and parallel) at a significantly higher accuracy (72.8% – *Casas* and 53.9% – *ExtraSensory*).

Dataset	**Ontological Approaches**	**Sequential Contexts**		**Parallel Contexts**		**Total Contexts**	
		Acc(JC)	Count	Acc(JC)	Count	Acc(JC)	Count
ExtraSensory [67]	MeetingMinds [70]	20.3%	976/4585	–	–/17265	0.03%	976/21850
	TAO (Ontology only)	**24.6%**	1138/4585	59%	10648/17265	**53.94%**	11786/21850
CASAS [21]	MeetingMinds [70]	11.03%	38949	*NA*	*NA*	11.03%	38949/182604
	TAO (Ontology only)	**72.80%**	153022/182604	*NA*	*NA*	**72.80%**	153022/182604

6.2.1 Ontological Pipeline. We evaluate the TAO system's ontological pipeline in terms of accuracy and richness in capturing contexts present in the two HAR datasets with rich activity patterns. We then compare our ontological approach with prior ontological approaches, such as the Meeting Minds (MM) context ontology [70] which models context as the relationship between activities. For a fair comparison, we update the activity labels from the datasets to be consistent with the activity instance definitions in the respective ontologies such that all the activities and their corresponding timestamps can be used for querying the ontology. Then, these activity labels



(a) Accuracy (JC) comparison (b) Precision comparison (c) Recall comparison

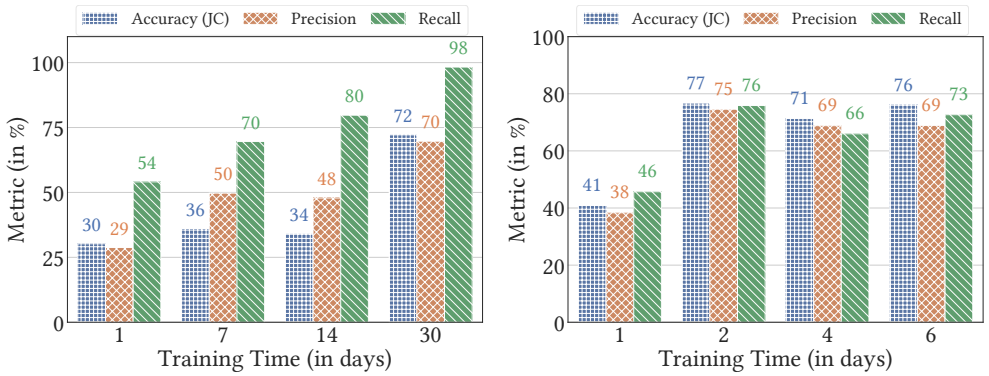
Fig. 10. Comparing the accuracy (JC), precision, and recall of context detection for TAO's temporal pipeline for different duration of *Lag Window*, across the *Casas* and the *Extrasensory* dataset. Our approach has high accuracy(65%-75%) for short (5 and 10 minutes) *Lag Window* in comparison to long (30 and 60 minutes) *Lag Window*. We also observed for both datasets, recall of context detection is consistently higher than the precision of context detection.

are posed to the respective reasoners, and the corresponding high-level contexts are inferred. Table 1 compares the accuracy (measured by the *Jaccard Similarity Coefficient* (JC)) and the contexts detected by each approach, categorized into sequential and parallel contacts. We observe that for both datasets, our approach detects contexts at a higher accuracy (*Extrasensory* - 53.94%, *Casas* - 72.8%) when compared to the MM approach (*Extrasensory* - 0.03%, *Casas* - 11.03%). We further observe that in the *Extrasensory* dataset, the low accuracy of the MM approach is primarily due to its inability of ontology to infer contexts from activity patterns that are parallel in nature (e.g., *A: talking*, *A: eating*, *A: tv* at the same time). In comparison, our approach is able to detect such contexts with high accuracy(60%). In addition, we observe that in the CASAS dataset, which consists of sequential activity patterns, our approach still outperforms MM (TAO 72.08% vs MM – 11.03%). This is because our approach consists of a denser vocabulary of activity relationships and contexts compared with prior work. Overall, we show that our ontological approach performs much better than prior approaches to detect an accurate and richer set of contexts.

6.2.2 Temporal Pipeline. We evaluate the performance of TAO's temporal pipeline across two key metrics, (1) *lag window*, which is the duration of the activity stream used for creating a sparse context representation(X_T), and (2) the amount of training data used for learning context clusters.

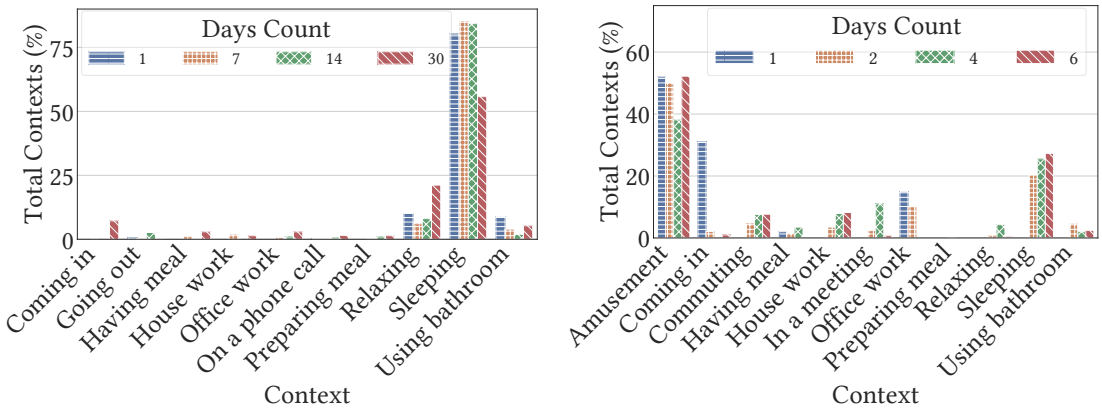
Optimizing the Lag Parameter: Figure 10 shows the different performance metrics of models trained on the *ExtraSensory* and the *Casas* datasets for different *lag parameters*: 5 min, 10 min, 30 min, and 60 min respectively. We see that accuracy (JC) decreases across both datasets as we increase the lag parameter. This shows that a lag parameter of 5 minutes is sufficient to capture various activity patterns happening across multiple contexts. Larger values (10min, 30min, 60mins) miss shorter duration contexts (i.e., *C: Using bathroom* between *C: Office work*) or overestimate context presence across a larger window (i.e., *C: Office work* and *C: Having meal* lasting for the entire 60-min window). We see a similar trend for precision and recall values, i.e., they decrease as the lag parameter increases. We see more dips in precision than recall, highlighting that shorter duration contexts are overestimated with larger lag values rather than shorter duration contexts being missed. We set the lag parameter to 5 mins based on our findings to provide the most accurate contexts for all subsequent evaluations.

Accuracy with Incremental Data: Figure 11 shows how the performance measures of our temporal pipeline change as the data used for training is increased, expressed as the number of days. To show this, we use data from two users chosen randomly , one from each dataset. The user from the *Casas* dataset has 45 days of ground truth data. Thus we trained different models with 1-day, 7 days, 14 days, and 30 days of training data. All trained models are then tested on the same one week of data to maintain consistency across users, which is not included in any training data. We use a similar protocol for the data from the user in the *ExtraSensory* dataset, except we trained different models with 1 day, 2 days, 4 days, and 6 days, and tested it on the same remaining two days in the end. We observe that for the *Casas* dataset user, we see a consistent increase in recall of context detection as training data increases. We get a good recall for contexts for a single day of training. However, the precision of



(a) Casas Dataset (b) Extrasensory Dataset

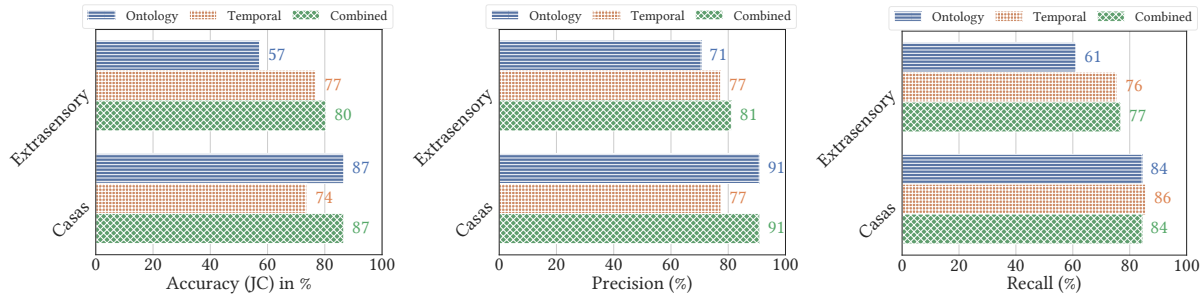
Fig. 11. Comparing the accuracy (JC), precision, and recall of context detection for TAO's temporal pipeline as the data (in days) available for training for a single user increases. For the *Casas* dataset, our approach shows a consistent increase in accuracy, precision, and recall values as the training data increases. For the *Extrasensory* dataset, our approach shows high accuracy within two days of training data.



(a) Casas Dataset Context Distribution (b) Extrasensory Dataset Context Distribution

Fig. 12. Comparing the distribution of predicted contexts learned over 45 days (1/7/14/30/45 days) for Casas and six days (1/2/4/6 days) for Extrasensory. In Casas, we see that with 30 days of training data, our pipeline can detect a wide range of contexts more accurately (72% JC). For the *Extrasensory* dataset, our approach shows that it can detect several contexts within two days of training data

context detection is low. For the user from the *ExtraSensory* dataset, we observe that precision and recall improve as we go from 1 to 2 days of training and then remain around the same as additional training data are used. The small dip in performance is an artifact of relatively less testing data. Figure 12 shows the distribution of contexts detected from the activity patterns. We see that in Casas, which predominantly consists of sequential activity patterns, with 30 days of training data, our pipeline can detect a wide range of contexts more accurately (72% JC). The context that is majorly detected is *C: Sleeping* since this activity happens *A: laying down, A: sleeping on bed* for a major period of the day. For the *Extrasensory* dataset, our approach can detect several contexts within two days of training data ranging from Amusement to Office Work.



(a) Accuracy (JC) comparison (b) Precision comparison (c) Recall comparison

Fig. 13. Comparing the accuracy (JC), precision, and recall of our TAO pipeline. We show results for the ontological pipeline only, the temporal pipeline only, and finally TAO's hybrid combined approach. Our temporal pipeline shows good accuracy across both datasets' accuracy (~75%). Our ontological pipeline shows higher accuracy (87%) in comparison to the temporal pipeline on the *Casas* dataset and a lower accuracy (57%) on the *Extrasensory* dataset. TAO's combined approach shows high accuracy (80% and 87%) and high precision(81% and 91%) for context detection across both datasets.

Overall Performance: Figure 13 shows the overall performance of our ontological approach and the temporal approach in isolation and then TAO's hybrid approach that combines them. For this evaluation, we split users into training and testing sets and trained one model per dataset. For the *Casas* dataset, we trained on data from 28 out of 30 users and then evaluated the remaining 2 users. For the *ExtraSensory* dataset, we used 53 out of 60 users for training and tested on the remaining 7 users. We observe that the temporal pipeline by itself has around 75% accuracy (JC) across both datasets. However, the ontological approach has significantly higher accuracy for the *Casas* dataset than the *ExtraSensory* dataset. This is attributable to the two datasets' different types of activity patterns. In *Casas*, most of the activities happen sequentially (i.e., *A: toilet>A: eating>A: washing dishes*), which leads to simpler mappings to context (i.e., *C: Using bathroom>C: Having meal>C: House work* etc.). However, in the *ExtraSensory* dataset, most activity patterns are interleaved for a long period of time (i.e., *A: walking>A: with friends+A: talking>A: talking*) and users are in multiple contexts simultaneously (i.e., *C: Commuting, C: Amusement* etc.). Due to this, our ontological only approach could not detect these contexts accurately. However, TAO's temporal pipeline tries to learn recurring patterns in the datasets instead of using a static mapping, thus can reason well about these patterns. Our combined approach performs better than individual components for both datasets. For the *Casas* dataset, our combined approach performs as well as the ontology pipeline as most activity patterns are sequential; thus, most of the time, the combined pipeline relies on output from ontology predictions. For *Extrasensory* dataset, our combined pipeline performs slightly better (80% vs. 77%) when compared with the temporal pipeline by itself. This improvement in performance can be attributed to relying on a temporal pipeline when there are recurring patterns of multi-context setting and using ontological predictions for more deterministic prediction where temporal clustering occasionally maps single activities to wrong context clusters.

6.3 *RQ2: Comparison of TAO's Performance with Prior Approaches*

While none of the prior approaches model the breadth of contexts as TAO does, there are some approaches, namely those proposed by Meditskos and Kompatsiaris [48] and Riboni et al. [60], that models a subset of the context we have and has a similar approach to TAO's ontological pipeline. Similar to TAO, both these research efforts [48, 60] focus on interleaved activities and use ontologies or probabilistic approaches to model them as contexts. In addition, both these approaches evaluate their approaches with a subset of activities from the *Casas* dataset [21] (which we use for our evaluations). Thus, we compare the performance of TAO with these approaches using the context-level accuracy for the subset of context that the approaches detect. It should be

Table 2. Accuracy comparison (F1 score) of TAO's combined pipeline with prior approaches, by Meditskos et al.[48] and Riboni et al. [60], to accurately detect context based on the *Casas* dataset [21]. Overall, TAO performs significantly better than these two approaches.

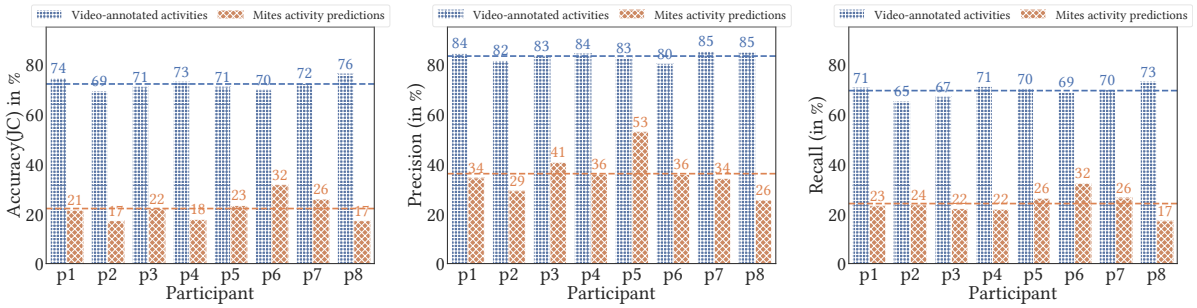
Dataset	Context	TAO	Meditskos et al. [48]	Riboni et al. [60]
Casas	*C: Coming in*	**46.5%**	-	-
	C: Going out	**52.9%**	-	-
	C: Having meal	**73.2%**	-	-
	C: House work	**87.7%**	81.8%	57.4%
	C: Office work	**89.1%**	-	-
	C: On a Phone call	55.8%	54%	**72.3%**
	C: Preparing meal	**88.4%**	78.4%	82.2%
	C: Relaxing	**86.4%**	79.7%	81.1%
	C: Sleeping	**82.4%**	-	-
	C: Using bathroom	**92.2%**	90.1%	88.2%

noted that both approaches [48, 60] use activity event information such as *A: picking spoon*, *A: teacup moved* to infer context such as *C: Having breakfast*. In contrast, TAO's hybrid approach only utilizes activities like *A: drinking*, *A: eating*, and models temporal patterns to infer a context such as *C: Having meal*. This limits our ability to make a direct comparison with prior approaches. Thus, we focus on how these methods compare with TAO in inferring different contexts on average. Table 2 shows a context-level F1-score of TAO and the performance of the corresponding context from both these approaches. For a fair comparison between these approaches, we ensure that the F1-score metric computed is done in the same way as reported in their respective papers. We observe that the TAO is not only able to detect a wide range of contexts when compared to the prior work, but we are also able accurately most of the contexts such as *C: Preparing meal*, *C: Relaxing* and *C: Using bathroom* with higher accuracy (> 85 F1-score), while other contexts such *C: House work* and *C: Phone call* with comparable accuracy.

6.4 *RQ3: Evaluation of TAO's Performance in Real World*

We conducted a real-world user study to evaluate the robustness of our TAO system for context detection from real-world activity patterns, beyond just activity datasets. In addition, we evaluate the accuracy of an underlying state-of-the-art activity detection pipeline using multi-modal sensors to characterize TAO's system performance.

6.4.1 Scenarios Definition. We define four scenarios that indicate common activity patterns that occur in daily life. In scenario 1, we define sequential activity patterns that indicate a context, such as *C: Person Entering* context would require the *A: door knock* activity to be sequentially followed by *A: door open* activity. In scenario 2, we define activity patterns that may indicate that an individual may be in multiple contexts simultaneously. For example, identifying that an individual is in the context of *C: On a phone call* while *C: Preparing coffee* requires that activity transition from *A: phone ring* to *A: talking* has occurred, and the individual is *A: grinding coffee*. In scenarios 3 and 4, we define interleaved and parallel activity patterns to define a context. For example, when an individual is in the context of *C: Exercising*, the activities such *A: jumping* and *A: running* are interleaved. Similarly, an individual can be in the context of *C: Relaxing*, *C: Having meal* and *C: Office work* if the activity *A: watching tv* is happening in the background while the individual shifts between *A: chewing food* or *A: clicking the mouse*. To emulate a real-world setting, we only put constraints on the entire scenario's start and end time of the entire scenario. However, users can perform given activity patterns in any manner they see fit. As an example, for scenario 4, a user might use frequent transitions for *A: eating*, *A: mouse click* while *A: tv* turns on and off in the background, or they can run *A: tv* for the entire period and focus most of their time *A: eating*, and only brief amount of time doing *A: mouse click*. More details of these scenarios are presented in the appendix, Table 5).



(a) Accuracy (JC) comparison (b) Precision comparison (c) Recall comparison

Fig. 14. Comparing the accuracy (JC), precision, and recall of TAO's context detection in our real-world user study. Activity inferences of different participants are obtained from either the annotated video data or predictions from the Mites platform [9]. TAO detects contexts at a higher accuracy (JC - 72%, precision: 83% and recall: 69%) using video-annotated activities, while the accuracy score drops (JC - 22%, precision: 36% and recall: 24%) when using the activities from Mites platform since the Mites activity detection pipeline was itself inaccurate at times.

6.4.2 User Study. We recruited 8 participants (four male and four female) with an average of 23 years, ranging between 21 and 32 years. Our study participants consisted mostly of university students and staff. Our institution's IRB approved our user study. We capture two types of data to identify activities and ground truth for contexts: (1) raw video from a camera and (2) activity inferences from a state-of-the-art platform called the Mites [9], that we were able to get access to. The raw video is later annotated manually for activities and their associated context. In addition, to understand how an underlying IoT device-based activity detection pipeline impacts TAO's accuracy for detecting contexts, we use the activity inferences from a state-of-the-art multi-modal sensor called Mites [9]. The Mites device captures environmental data, such as ambient temperature, relative atmospheric humidity, pressure, vibrations, motion, light color, thermal temperature, and sound, and combines this information to perform activity recognition using an SVM-based ML model. The authors shared a set of sensors and their activity detection pipeline with us for our project. We train an ML model for each user based on the Mites sensor data when the user performs the specified activities. Once the model is trained and deployed for activity inference, TAO uses the provided activity predictions as input for its own context recognition.

As part of the study, we asked each participant to perform the $3-4$ different activity patterns as mentioned in each of the four scenarios (more details in appendix, Table 5). The study was performed in a conference room on our university campus, and the participants could perform the activities themselves as they saw fit. Each activity pattern was performed for a specific duration (4 to 5 mins), and the participants perform each activity pattern three times to simulate data collection over multiple time periods. We use the annotated activity information from the video data and the activity inferences from the Mites sensor's ML model to predict the context for each user using TAO's combined approach. To characterize TAO's performance in this real-world study, we ran a leave-one-participant-out (LOPO) cross-validation to measure accuracy (JC). Figure 14 shows LOPO performance for all 8 participants with two kinds of activity inputs, (i), where TAO takes activity inputs from video annotated data, and (ii), where TAO takes activity inputs from the Mites ML activity detection pipeline.

We observe that when TAO uses the annotated activities from the video data, it can detect contexts relatively accurately (JC - 72%, Precision: 83% and Recall: 69%), which is similar to the performance on the HAR datasets. In addition, we observe that when the TAO system obtains the activity input from the Mites activity recognition pipeline TAO's context detection accuracy score drops to (JC - 22%, Precision: 36% and Recall: 24%) for the same participant data. To understand the drop in accuracy when using the activities detected by the Mites ML pipeline, we evaluate the precision and recall at each context level averaged across all participants as shown in Table 3.

Table 3. Context level performance for the real world dataset

Context	Video-annotated activities		Mites activity predictions	
	Precision	Recall	Precision	Recall
Coming In	98.2%	99.68%	17.47%	58.04%
Exercising	97.56%	99.42%	33.99%	36.36%
Having Meal	87.29%	87.66%	16.57%	17.43%
House Work	88.35%	87.66%	23.48%	4.58%
Office Work	98.78%	73.04%	55.96%	36.02%
Phone Call	69.81%	39.98%	21.17%	13.36%
Relaxing	77.87%	58.04%	46.30%	24.44%

We see that for contexts such as *C: Coming in* or *C: On a phone call*, the precision and recall values decreased compared with video annotated activities, showing that the activity prediction from the Mites detection pipeline is itself not accurate. With additional training data and better sensing approaches, one can achieve higher activity recognition accuracy, leading to better context detection accuracy with TAO. We note that detecting human activities using different sensor modalities and advanced machine learning techniques is an active area of research, and any advances made in that domain would only help increase TAO's accuracy even further.

7 APPLICATION: CONTEXT TO WELLNESS PIPELINE

This section presents two exemplary applications that can use rich contextual information for TAO to understand useful wellness metrics for users.

7.1 Wellness Metrics for Productivity

Prior work [11, 66] has shown that an individual's brain can only focus for 90 to 120 minutes, at which point it needs a short break before launching into the next 90– to 120–minute period of focus. This cycle is known as an *ultradian* rhythm; therefore, it is important for an individual to learn their rhythm or routine to maximize their productivity. Several factors can affect productivity, including personality traits such as procrastination, distractions, multitasking [12], etc. The most common strategies proposed to improve productivity fall into three main categories [27]:

- 15-minute breaks [63]: Plan multiple 15-minute breaks during your eight-hour workday to break up long work stretches.
- The Pomodoro Technique [18]: this technique involves breaking your day into half-hour segments called *pomodoros* that include 25 minutes of focus followed by 5 minutes of rest. Complete four pomodoros, then take a 15-20 minute break.
- 90-minute windows: 90-minute work windows for a single task and then taking a 20-minute break before your next 90-minute window.

Thus, we build our application based on these metrics and use the contexts derived from TAO to identify how long an individual remains in the *C: OfficeWork* context and if they switch contexts to ones that denote a break (e.g., *C: Relaxing*) at appropriate intervals. We also assign a productivity score based on the number of times the individual failed to follow these metrics (e.g., a negative score when the user is continuously in an office work context without taking a break for an hour). Based on these factors, we illustrate how the productivity scores of different users from the *Extrasensory* dataset change over a week in Figure 15. We observe that user 1 shows lower productivity for the middle part of the week compared to the start and the end of the week. In contrast, the productivity of user 6 remains low over the entire week. This example showcases the potential of

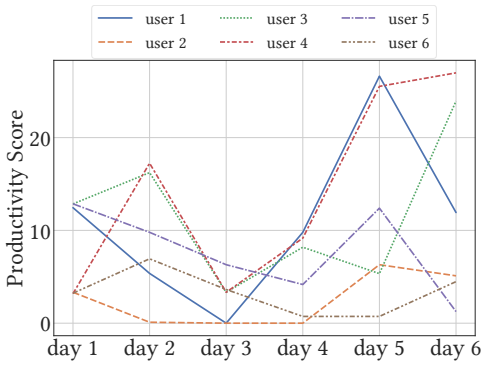


Fig. 15. Productivity score for six users from the *Extrasensory* dataset over six days.

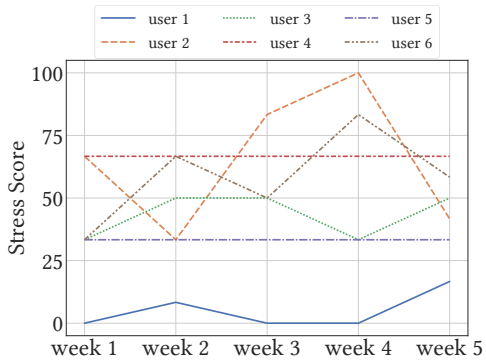


Fig. 16. Stress score for six users from the *Casas* dataset over five weeks.

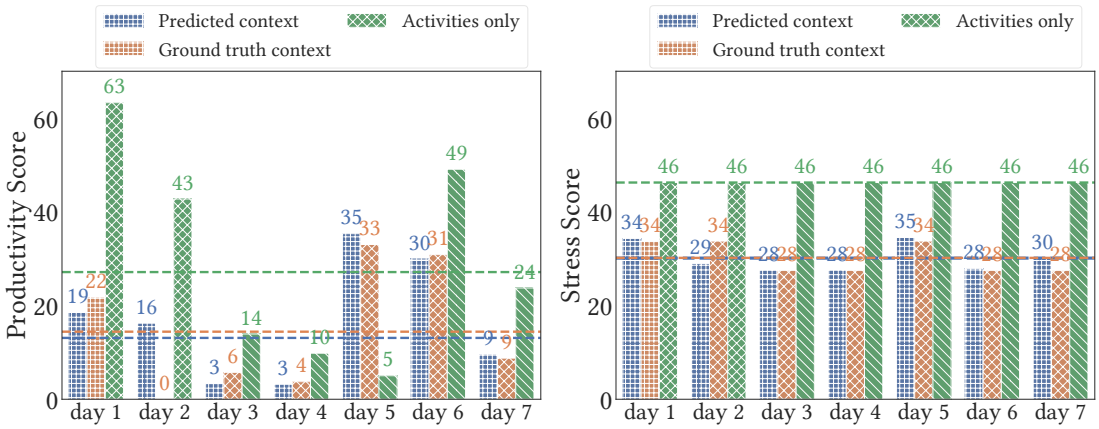
using context predictions from TAO to understand productivity. In the future, an interesting application can be built by performing a more detailed analysis of the context patterns leading to changes in productivity.

7.2 Wellness Metrics for Stress

Several survey-based studies show a direct correlation between physiological stress and lifestyle based on data from long-term behavior patterns. These long-term patterns can be expressed in terms of different contexts a person is in, the duration of time a person spends in a given context, and the number of times they are in a context within a day or a week. We summarize the findings from 11 surveys across a wide variety of demographics where context patterns are associated with varying stress levels for an individual:

- Absence of physical activity: *Cao et al.* [14] study the correlation between physical activity and perceived stress and show that *"no physical activity and low intensity of physical activities are associated with risk of high stress"* (i.e., absence of *C: Exercising*, and *C: House work* contexts).
- Impact of transport-related activities: Cao et al. [14] also shows that *high-intensity of transport activities lead to high-stress"* (i.e., daily patterns of *C: Commuting* for an individual user).
- Patterns of working hours: Can et al. [13] study the impact of working patterns and physical activity together on stress and shows that *"individuals who work shifts and do physical activities regularly have moderate stress, individuals who work shifts but do not do physical activity have high-stress, and those who do regular exercise and work on a regular day shift have a lower stress score"* (i.e. Weekly patterns for *C: Office work*, and *C: Exercising*). Some studies quantify how many working hours are too long and can lead to high stress for an individual [73]. They show that *working more than 10 hours/day or 50 hours/week leads to a high risk of experiencing occupational health problems.*
- Weekend behavior patterns: Some studies distinguish between weekday and weekend working behavior [62] and find that *the negative effect of an hour increase in weekend work is one and a half to two times larger than that of weekday overtime work for white-collar workers.*. They report other findings, including *taking a relatively long rest on weekends is key for reducing stress*, and *working after midnight is associated with mental health issues* (i.e. Contexts patterns for *C: Relaxing*, and *C: Office work* on weekends).

We prototyped an application to identify these context patterns, which can (potentially) relate to increasing or decreasing an individual's stress level. Our aim is not to quantify an individual's stress levels, as it is multi-faceted and can be subjective. However, keeping track of these context patterns can potentially help provide personalized recommendations for improving the wellness of an individual. We assigned all the markers a score based on



(a) Productivity score comparison

(b) Stress score comparison

Fig. 17. Comparison of productivity and stress scores generated using context prediction from TAO pipeline, ground truth context labels, and only using the activity information for a single user (user 3) in the *Extrasensory* dataset. Figure 17(a) and 17(b) shows that both the productivity scores (avg/week = 16.57) and stress scores (avg/week = 30.12) generated from TAO's context prediction are similar to the ground truth productivity (avg/week = 14.41) and stress score (avg/week = 30.17), while the productivity (avg/week = 29.65) and stress (avg/week = 46.25) scores generated from the activity information are different.

severity (low or high) and impact (increasing or decreasing) on stress reported in survey-based studies. We approximated individuals' weekly and daily stress scores based on the cumulative score of these context patterns identified in the week. Figure 16 shows a variation in the weekly stress score of six users from the *Casas* dataset. We see a wide variety of stress patterns across the week. User 1, user 5, and user 4 show consistent stress scores over weeks with low, medium, and high-stress scores, respectively, which indicates there is not much change in the lifestyle of these users across weeks. User 2 shows a sudden increase in their stress score, which might indicate a sudden change in lifestyle toward higher stress. User 6 shows an increasing stress score over time, and user 3 shows an increase and a decrease in stress score over consecutive weeks. These patterns can be studied further by identifying the context patterns leading to these scores.

7.3 Significance of Context Recognition from TAO for Wellness Applications

In both the wellness applications discussed, the context inference from the TAO system plays a crucial role in calculating the productivity and stress score values. To illustrate this, in Figure 17, we show the comparison of productivity and stress scores generated with and without using TAO's context predictions, i.e., only use the activity inferences and with the manually annotated ground truth context predictions. We pick the activity information for one of the two users we annotated the ground truth for (user 3 in Figure 15) in the Extrasensory dataset and compute the productivity and stress scores for a week. For a fair comparison, when we provide activities as input to calculate the productivity or stress scores, we create a direct mapping of the activities to possible office context, e.g., we map multiple activities such as A: computer work, A: typing to a context such as *C: Office work* to emulate what a real-world user would do. We provide these as input to the productivity and stress application and compare their scores. We observe in Figure 17(a) that the productivity score generated from the TAO's predicted contexts (average/week = 16.57) is very similar to the scores generated from the manually annotated ground truth context information (average/week = 14.41) indicating the user had lower productivity over the week. However, when we directly use activity inference to calculate the productivity score, we obtain a

high productivity score (average/week = 29.65)), when in comparison to ground truth, the individual did indeed have lower productivity. Similarly, in Figure 17(b), we see that the TAO's context predictions (average/week = 30.12) provide similar stress scores to that of the ground truth contexts (average/week = 30.17), while only using this activity information generates higher stress scores (average/week = 46). These results show that the accurate context predictions from TAO are crucial for providing meaningful metrics for wellness-based applications reporting occupant productivity or stress.

More importantly, these applications are merely examples of the ones that can be built using the rich contexts detected from our TAO system. Importantly, there is no reason to stop at these applications; a wide range of application scenarios can utilize contextual information, for example, understanding eating habits, detecting sudden lifestyle changes, monitoring the elderly in AAL environments, etc.

8 DISCUSSION AND LIMITATIONS

Context awareness has long been proposed as a way to allow computing technologies to understand human behavior and the environment, using it to adapt and provide meaningful insights to users dynamically.. Using IoT sensors, coupled with machine learning to sense human activities and then, in turn, detect semantically meaningful contexts is the natural next step. Through our TAO system, we have shown that understanding the context from complex activity patterns is indeed feasible and can enable powerful applications. Here we discuss some of the limitations and further advances needed for even better context detection.

Improvements to the Underlying Sensing Approaches: The sensing technologies we have today, which include numerous commercially-available physical devices (e.g., smart cameras, multi-modal devices) coupled with ML algorithms for processing this sensor data—are already enabling a richer understanding of people's everyday activities. However, many of these techniques are still evaluated in highly controlled settings and still report HAR accuracy of 70%, 80% [8, 40]. While promising, there is still research that is needed to explore more advanced sensing modalities and activity sensing pipelines that do not need a significant amount of in-situ labeled training data while still being accurate (e.g. 90% to 95%) in real-world settings.

New Tools and Methods for End-Users or Designers to Encode New Contexts: While TAO can detect meaningful contexts from parallel, sequential, or interleaved activities, the majority of the contexts modeled are based on the ontology we have created ourselves. Given that new contexts will likely be discovered and will require extending the ontology, we need tools for users and designers to express their context definition, such as activity-context rules, etc. [43].

Rationale to Use Jaccard Coefficient as an Accuracy Metric: We have designed TAO to utilize the temporal and ontological pipelines and provide the most probable context given an activity pattern. However, in the real world annotating ground truth with all the contexts possible is quite challenging and thus often limited to annotating a single context for different time extents. This can result in the context detected by the TAO pipeline not always being a direct match. Thus, it is important to have a performance metric that considers partial context detection, even if it is not an exact match. One way to accomplish this is to give an accuracy score based on the fraction of ground truth identified by the context detection system. However, in that case, a system that predicts all the contexts for every request would have 100% accuracy, which is incorrect. *Jaccard Similarity Coefficient* is an appropriate metric to determine the efficacy of these scenarios as it takes into account the case when a subset of ground truth contexts are detected while also penalizing detecting additional contexts that are incorrect.

Scenarios when TAO Misses Contexts: We show that TAO system can detect contexts with an accuracy (in terms of average *Jaccard Similarity Coefficient*) of 80% and 87% for large-scale real-world activity recognition datasets, *ExtraSensory*[67], and *Casas*[21] respectively. Our approach does better than other approaches in terms of the richness of contexts and is at par regarding the precision and recall of most contexts. However, there are still opportunities to improve TAO's accuracy further. Some of the sources of inaccuracy include:

- **Transition between contexts:** learning activity patterns that occur when users transition between two contexts (i.e., a user might transition from *A: sleeping>A: taking a shower* every morning, which is a common pattern but does not represent any meaningful context). This is particularly problematic for shorter contexts like *C: Using bathroom*, *C: Coming in*, or *C: Going out*.
- **Combining individual components:** Currently, we are using a naive approach to combine contexts detected from our ontological and temporal pipeline by taking a union of contexts from both pipelines. This provides a richer set of contexts leading to higher recall values for context detection. It also leads to a loss in precision when either of the pipelines detects an unnecessary context. A better way to combine both approaches could utilize the characteristics of activity traces to dynamically make a decision of using one of the pipelines preferentially.

Sensitivity to Underlying Activity Detection Pipeline: In our real-world study, we observed that our accuracy (JC) drops significantly due to errors in the underlying activity detection pipeline. One way to address this issue is to choose an architecture that is robust to noise in training data by using an ensemble of deep learning models that trains on noisy data. However, different kinds of activity detection models can introduce different kinds of noise, i.e., some could miss actual activity inferences, whereas others can predict inaccurate activities due to noise in sensor data. It is a challenging problem to create a system that is robust against inaccurate activity prediction.

9 CONCLUSION

In this paper, we present TAO, a context detection system that combines ontological and deep unsupervised clustering approaches for inferring a rich set of contexts from a wide variety of daily activities. The TAO system models the different activity patterns sequential, parallel, or interleaved activities as context information using the OWL-based ontologies. The temporal pipeline uses an unsupervised clustering algorithm to detect context from new activity patterns and automatically extends our ontology based on new activity patterns. Our system is agnostic to the underlying activity detection mechanism and set of activities detected, making it usable across various sensing systems to derive semantically richer information from activity streams. We showed that the TAO system performs well across multiple settings we explored using two well-known public datasets and our real-world study. The TAO system is an end-to-end system, which can be used right out of the box with activity streams from daily living activities, and starts detecting context accurately with a few days of training data. We prototype two exemplary applications that use rich contextual information from the TAO system to provide wellness metrics to users towards understanding productivity and indicators of stress.

ACKNOWLEDGMENTS

This work was partially supported by NSF Award SaTC-1801472 and the CMU's CyLab Security and Privacy Institute. We gratefully acknowledge a gift by JP Morgan Chase to support research on smart buildings at Carnegie Mellon. We want to thank Mayank Goel, Catherine Tianhong Yu, Neeha Dev Arun, and Vimal Mollyn for their invaluable feedback on the early revisions of the paper. We also thank our anonymous reviewers for their constructive feedback on our paper.

REFERENCES

[1] Alireza Abdoli. 2021. Time Series Data Mining Algorithms Towards Scalable and Real-Time Behavior Monitoring. https://doi.org/10.48550/ARXIV.2112.14630

[2] Alessandra Agostini, Claudio Bettini, and Daniele Riboni. 2009. Hybrid reasoning in the CARE middleware for context awareness. *International journal of Web engineering and technology* 5, 1 (2009), 3–23.

[3] Jose Aguilar, Marxjhony Jerez, and Taniana Rodríguez. 2018. CAMeOnto: Context awareness meta ontology modeling. *Applied computing and informatics* 14, 2 (2018), 202–213.

[4] Ane Alberdi, Asier Aztiria, and Adrian Basarab. 2016. Towards an automatic early stress recognition system for office environments based on multimodal measurements: A review. *Journal of biomedical informatics* 59 (2016), 49–75.

[5] Louis Atallah and Guang-Zhong Yang. 2009. The use of pervasive sensing for behaviour profiling—a survey. *Pervasive and mobile computing* 5, 5 (2009), 447–464.

[6] Gustavo E. Batista, Eamonn J. Keogh, Oben Moses Tataw, and Vinícius M. Souza. 2014. CID: An Efficient Complexity-Invariant Distance for Time Series. *Data Min. Knowl. Discov.* 28, 3 (may 2014), 634–669. https://doi.org/10.1007/s10618-013-0312-3

[7] Claudio Bettini, Oliver Brdiczka, Karen Henricksen, Jadwiga Indulska, Daniela Nicklas, Anand Ranganathan, and Daniele Riboni. 2010. A survey of context modelling and reasoning techniques. *Pervasive and mobile computing* 6, 2 (2010), 161–180.

[8] Sejal Bhalla, Mayank Goel, and Rushil Khurana. 2022. IMU2Doppler: Cross-Modal Domain Adaptation for Doppler-Based Activity Recognition Using IMU Data. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 4, Article 145 (dec 2022), 20 pages. https://doi.org/10.1145/3494994

[9] Sudershan Boovaraghavan, Chen Chen, Anurag Maravi, Mike Czapik, Yang Zhang, Chris Harrison, and Yuvraj Agarwal. 2023. Mites: Design and Deployment of a General-Purpose Sensing Infrastructure for Buildings. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 1, Article 2 (mar 2023), 32 pages. https://doi.org/10.1145/3580865

[10] Sudershan Boovaraghavan, Anurag Maravi, Prahaladha Mallela, and Yuvraj Agarwal. 2021. MLIoT: An End-to-End Machine Learning System for the Internet-of-Things. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation* (Charlottesvle, VA, USA) *(IoTDI '21)*. Association for Computing Machinery, New York, NY, USA, 169–181. https://doi.org/10.1145/3450268.3453522

[11] Aisha Bradshaw. 2019. Wellness in the workplace. *Nature Human Behaviour* 3, 10 (2019), 1039–1039.

[12] Business News Daily. 2023. Biggest Workplace Distractions That Kill Productivity - businessnewsdaily.com. https://www.businessnewsdaily.com/8098-distractions-kiling-productivity.html.

[13] Sema Can. 2019. The Determining of Relationship between Physical Activity and Perceived Stress Level in Security Service Employees. *Journal of Education and Training Studies* 7, 1 (2019), 149.

[14] Bing Cao, Yuxiao Zhao, Zhongyu Ren, Roger S McIntyre, Kayla M Teopiz, Xiao Gao, and Ling Ding. 2021. Are physical activities associated with perceived stress? The evidence from the China Health and Nutrition Survey. *Frontiers in public health* 9 (2021), 1104.

[15] Harry Chen, Tim Finin, and Anupam Joshi. 2005. The SOUPA Ontology for Pervasive Computing. In *Ontologies for Agents: Theory and Experiences*. Birkhäuser Basel, Basel, 233–258.

[16] Luke Chen and CD Nugent. 2009. Ontology-based activity recognition in intelligent pervasive environments. *International Journal of Web Information Systems* 5, 4 (2009), 410–430. https://doi.org/10.1108/17440080911006199

[17] Prerna Chikersal et al. 2021. Detecting depression and predicting its onset using longitudinal symptoms captured by passive sensing: a machine learning approach with robust feature selection. *ACM Transactions on Computer-Human Interaction (TOCHI)* 28, 1 (2021), 1–41.

[18] Cirillo Consulting GmbH. 2023. The Pomodoro® Technique | Cirillo Consulting GmbH. https://pomodorotechnique.com/.

[19] Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. 2020. Leveraging Procedural Generation to Benchmark Reinforcement Learning. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, online, 2048–2056. https://proceedings.mlr.press/v119/cobbe20a.html

[20] Diane J Cook, Juan C Augusto, and Vikramaditya R Jakkula. 2009. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing* 5, 4 (2009), 277–298.

[21] Diane J Cook, Aaron S Crandall, Brian L Thomas, and Narayanan C Krishnan. 2012. CASAS: A smart home in a box. *Computer* 46, 7 (2012), 62–69.

[22] Antonia Creswell, Kai Arulkumaran, and Anil A. Bharath. 2017. On denoising autoencoders trained to minimise binary cross-entropy. arXiv:1708.08487 [cs.CV]

[23] Ehsan Elhamifar and Dat Huynh. 2020. Self-supervised Multi-task Procedure Learning from Instructional Videos. In *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 557–573.

[24] Karan Goel and Emma Brunskill. 2019. Learning Procedural Abstractions and Evaluating Discrete Latent Temporal Structure. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, US, 10. https://openreview.net/forum?id=ByleB2CcKm

[25] Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter Patel-Schneider, and Ulrike Sattler. 2008. OWL 2: The next step for OWL. *Journal of Web Semantics* 6, 4 (2008), 309–322.

[26] Andrew Greasley and Chris Owen. 2016. *Behavior in Models: A Framework for Representing Human Behavior*. Palgrave Macmillan UK, London, 47–63. https://doi.org/10.1057/978-1-137-53551-1_3

[27] Barry P Haynes. 2007. Office productivity: a shift from cost reduction to human contribution.

[28] Rim Helaoui, Daniele Riboni, and Heiner Stuckenschmidt. 2013. A Probabilistic Ontological Framework for the Recognition of Multilevel Human Activities. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Zurich, Switzerland) *(UbiComp '13)*. Association for Computing Machinery, New York, NY, USA, 345–354. https://doi.org/10.1145/2493432.

2493501
[29] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation* 18, 7 (2006), 1527–1554.
[30] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
[31] Jin-Hyuk Hong, Julian Ramos, and Anind K. Dey. 2016. Toward Personalized Activity Recognition Systems With a Semipopulation Approach. *IEEE Transactions on Human-Machine Systems* 46, 1 (2016), 101–112. https://doi.org/10.1109/THMS.2015.2489688
[32] Walaa N. Ismail, Mohammad Mehedi Hassan, and Hessah A. Alsalamah. 2019. Context-Enriched Regular Human Behavioral Pattern Detection From Body Sensors Data. *IEEE Access* 7 (2019), 33834–33850. https://doi.org/10.1109/ACCESS.2019.2904122
[33] Anil K. Jain and Richard C. Dubes. 1988. *Algorithms for Clustering Data.* Prentice-Hall, Inc., USA.
[34] Rohan Kabra, Divya Saxena, Dhaval Patel, and Jiannong Cao. 2021. Time Series Clustering for Human Behavior Pattern Mining. https://doi.org/10.48550/ARXIV.2110.07549
[35] Sabin Kafle and Dejing Dou. 2016. A Heterogeneous Clustering Approach for Human Activity Recognition. In *Big Data Analytics and Knowledge Discovery*, Sanjay Madria and Takahiro Hara (Eds.). Springer International Publishing, Cham, 68–81.
[36] Harmanpreet Kaur, Alex C. Williams, Daniel McDuff, Mary Czerwinski, Jaime Teevan, and Shamsi T. Iqbal. 2020. Optimizing for Happiness and Productivity: Modeling Opportune Moments for Transitions and Breaks at Work. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3313831.3376817
[37] Nicky Kern, Bernt Schiele, and Albrecht Schmidt. 2003. Multi-sensor Activity Context Detection for Wearable Computing. In *Ambient Intelligence*, Emile Aarts, René W. Collier, Evert van Loenen, and Boris de Ruyter (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 220–232.
[38] Sunder Ali Khowaja, Aria Ghora Prabono, Feri Setiawan, Bernardo Nugroho Yahya, and Seok-Lyong Lee. 2018. Contextual activity based Healthcare Internet of Things, Services, and People (HIoTSP): An architectural framework for healthcare monitoring using wearable sensors. *Computer Networks* 145 (2018), 190–206.
[39] Hilde Kuehne, Ali Arslan, and Thomas Serre. 2014. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, US, 780–787.
[40] Gierad Laput, Karan Ahuja, Mayank Goel, and Chris Harrison. 2018. Ubicoustics: Plug-and-Play Acoustic Activity Recognition. In *Proc. of the 31st Annual ACM Symposium on UIST* (Berlin, Germany) *(UIST '18)*. ACM, New York, NY, USA, 213–224. https://doi.org/10.1145/3242587.3242609
[41] Kang Li and Yun Fu. 2014. Prediction of Human Activity by Discovering Temporal Sequence Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 8 (2014), 1644–1657. https://doi.org/10.1109/TPAMI.2013.2297321
[42] Li Liu, Yuxin Peng, Ming Liu, and Zigang Huang. 2015. Sensor-based human activity recognition system with a multilayered model using time series shapelets. *Knowledge-Based Systems* 90 (2015), 138–152.
[43] Ryan Louie, Darren Gergle, and Haoqi Zhang. 2022. Affinder: Expressing Concepts of Situations That Afford Activities Using Context-Detectors. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 7, 18 pages. https://doi.org/10.1145/3491102.3501902
[44] Dimitrios Lymberopoulos, Athanasios Bamis, and Andreas Savvides. 2008. Extracting Spatiotemporal Human Activity Patterns in Assisted Living Using a Home Sensor Network. In *Proceedings of the 1st International Conference on PErvasive Technologies Related to Assistive Environments* (Athens, Greece) *(PETRA '08)*. Association for Computing Machinery, New York, NY, USA, Article 29, 8 pages. https://doi.org/10.1145/1389586.1389621
[45] Donald M MacKay. 1956. Towards an information-flow model of human behaviour. *British Journal of Psychology* 47, 1 (1956), 30–43.
[46] Naveen Sai Madiraju, Seid M. Sadat, Dimitry Fisher, and Homa Karimabadi. 2018. Deep Temporal Clustering : Fully Unsupervised Learning of Time-Domain Features. https://doi.org/10.48550/ARXIV.1802.01059
[47] Georgios Meditskos, Stamatia Dasiopoulou, and Ioannis Kompatsiaris. 2016. MetaQ: A knowledge-driven framework for context-aware activity recognition combining SPARQL and OWL 2 activity patterns. *Pervasive and Mobile Computing* 25 (2016), 104–124.
[48] Georgios Meditskos and Ioannis Kompatsiaris. 2017. iKnow: Ontology-driven situational awareness for the recognition of activities of daily living. *Pervasive and Mobile Computing* 40 (2017), 17–41.
[49] Mehrab Bin Morshed, Koustuv Saha, Richard Li, Sidney K. D'Mello, Munmun De Choudhury, Gregory D. Abowd, and Thomas Plötz. 2019. Prediction of Mood Instability with Passive Sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 3, Article 75 (sep 2019), 21 pages. https://doi.org/10.1145/3351233
[50] Mark A. Musen. 2015. The protégé project: a look back and a look forward. *AI Matters* 1, 4 (2015), 4–12. https://doi.org/10.1145/2757001.2757003
[51] George Okeyo, Liming Chen, Hui Wang, and Roy Sterritt. 2011. Ontology-based learning framework for activity assistance in an adaptive smart home. In *Activity recognition in pervasive intelligent environments*. Springer, US, 237–263.

[52] Adam Paszke et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., US, 8024–8035. https://arxiv.org/abs/1912.01703
[53] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. 2013. Context aware computing for the internet of things: A survey. *IEEE communications surveys & tutorials* 16, 1 (2013), 414–454.
[54] Preeja Pradeep and Shivsubramani Krishnamoorthy. 2019. The MOM of context-aware systems: A survey. *Computer Communications* 137 (2019), 44–69.
[55] Reza Rawassizadeh, Elaheh Momeni, Chelsea Dobbins, Joobin Gharibshah, and Michael Pazzani. 2016. Scalable Daily Human Behavioral Pattern Mining from Multivariate Temporal Data. *IEEE Transactions on Knowledge and Data Engineering* 28, 11 (2016), 3098–3112. https://doi.org/10.1109/TKDE.2016.2592527
[56] Daniele Riboni and Claudio Bettini. 2011. COSAR: hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing* 15, 3 (2011), 271–289.
[57] Daniele Riboni and Claudio Bettini. 2011. OWL 2 modeling and reasoning with complex human activities. *Pervasive and Mobile Computing* 7, 3 (2011), 379–395.
[58] Daniele Riboni, Claudio Bettini, Gabriele Civitarese, Zaffar Haider Janjua, and Rim Helaoui. 2015. Fine-grained recognition of abnormal behaviors for early detection of mild cognitive impairment. In *2015 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, US, 149–154. https://doi.org/10.1109/PERCOM.2015.7146521
[59] Daniele Riboni, Claudio Bettini, Gabriele Civitarese, Zaffar Haider Janjua, and Rim Helaoui. 2016. SmartFABER: Recognizing fine-grained abnormal behaviors for early detection of mild cognitive impairment. *Artificial intelligence in medicine* 67 (2016), 57–74.
[60] Daniele Riboni, Timo Sztyler, Gabriele Civitarese, and Heiner Stuckenschmidt. 2016. Unsupervised Recognition of Interleaved Activities of Daily Living through Ontological and Probabilistic Reasoning. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Heidelberg, Germany) *(UbiComp '16)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/2971648.2971691
[61] S. Harris, A. Seaborne. 2023. SPARQL 1.1 query language, W3C recommendation. http://www.w3.org/TR/sparql11-query/.
[62] Kaori Sato, Sachiko Kuroda, and Hideo Owan. 2020. Mental health effects of long work hours, night and weekend work, and short rest periods. *Social Science & Medicine* 246 (2020), 112774.
[63] Scientific American. 2023. Happiness Is a Walk in the Park - Scientific American. https://www.scientificamerican.com/podcast/episode/happiness-is-a-walk-in-the-park-10-05-05/.
[64] Sudershan Boovaraghavan, Prasoon Patidar, Yuvraj Agarwal. 2023. TAO: Open-source repository for the TAO system. https://github.com/synergylabs/tao.
[65] RAPIDS Development Team. 2018. RAPIDS: Collection of Libraries for End to End GPU Data Science. https://rapids.ai
[66] Toggl. 2023. Toggl: Time Tracking, Project Planning and Hiring Tools to Help Teams Work BetterToggl: Time Tracking, Project Planning and Hiring Tools to Help Teams Work Better. https://toggl.com/.
[67] Yonatan Vaizman, Katherine Ellis, and Gert Lanckriet. 2017. Recognizing detailed human context in the wild from smartphones and smartwatches. *IEEE pervasive computing* 16, 4 (2017), 62–74.
[68] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html
[69] T. L. M. van Kasteren, G. Englebienne, and B. J. A. Kröse. 2011. *Human Activity Recognition from Wireless Sensor Network Data: Benchmark and Software*. Atlantis Press, Paris, 165–186. https://doi.org/10.2991/978-94-91216-05-3_8
[70] Claudia Villalonga, Muhammad Asif Razzaq, Wajahat Ali Khan, Hector Pomares, Ignacio Rojas, Sungyoung Lee, and Oresti Banos. 2016. Ontology-based high-level context inference for human behavior identification. *Sensors* 16, 10 (2016), 1617.
[71] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and Composing Robust Features with Denoising Autoencoders. In *Proceedings of the 25th International Conference on Machine Learning* (Helsinki, Finland) *(ICML '08)*. Association for Computing Machinery, New York, NY, USA, 1096–1103. https://doi.org/10.1145/1390156.1390294
[72] W3C. 2023. SPIN - SPARQL Inferencing Notation. https://spinrdf.org/.
[73] Kapo Wong, Alan HS Chan, and SC Ngan. 2019. The effect of long working hours and overtime on occupational health: a meta-analysis of evidence from 1998 to 2018. *International journal of environmental research and public health* 16, 12 (2019), 2102.
[74] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised Deep Embedding for Clustering Analysis. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48 (ICML'16)*. JMLR.org, New York, NY, USA, 478–487.
[75] Frank F. Xu, Lei Ji, Botian Shi, Junyi Du, Graham Neubig, Yonatan Bisk, and Nan Duan. 2020. A Benchmark for Structured Procedural Knowledge Extraction from Cooking Videos. https://doi.org/10.48550/ARXIV.2005.00706
[76] Juan Ye, Graeme Stevenson, and Simon Dobson. 2014. USMART: An unsupervised semantic mining activity recognition technique. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 4, 4 (2014), 1–27.
[77] Kristina Yordanova et al. 2019. Analysing Cooking Behaviour in Home Settings: Towards Health Monitoring. *Sensors (Basel, Switzerland)* 19, 3 (February 2019), E646. https://doi.org/10.3390/s19030646

[78] Chunhui Yuan and Haitao Yang. 2019. Research on K-Value Selection Method of K-Means Clustering Algorithm. *J* 2, 2 (2019), 226–235. https://doi.org/10.3390/j2020016

[79] Junhai Zhai, Sufang Zhang, Junfen Chen, and Qiang He. 2018. Autoencoder and Its Various Variants. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE Press, Miyazaki, Japan, 415–419. https://doi.org/10.1109/SMC.2018.00080

[80] Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew-Soon Ong, and Chen Change Loy. 2020. Online Deep Clustering for Unsupervised Representation Learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, online, 6687–6696. https://doi.org/10.1109/CVPR42600.2020.00672

[81] Jiangchuan Zheng and Lionel M. Ni. 2012. An Unsupervised Framework for Sensing Individual and Cluster Behavior Patterns from Human Mobile Data. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (Pittsburgh, Pennsylvania) *(UbiComp '12)*. Association for Computing Machinery, New York, NY, USA, 153–162. https://doi.org/10.1145/2370216.2370241

[82] Luowei Zhou, Chenliang Xu, and Jason Corso. 2018. Towards Automatic Learning of Procedures From Web Instructional Videos. *Proceedings of the AAAI Conference on Artificial Intelligence* 32, 1 (Apr. 2018), 1. https://doi.org/10.1609/aaai.v32i1.12342

A APPENDICES

In this section, we provide additional details for the papers, specifically the example use cases of TAO's context recognition from ontology and more details on the dataset characteristics and ground truth annotation.

A.1 Ontology Details

In this section, we delve into the detailed representation of the ontology utilized in our study. The ontology serves as a structured framework that defines the concepts, relationships, and hierarchies within a specific domain. By elucidating the ontology's structure, we aim to provide a comprehensive understanding of how context and activity information are organized and interconnected using concrete examples.

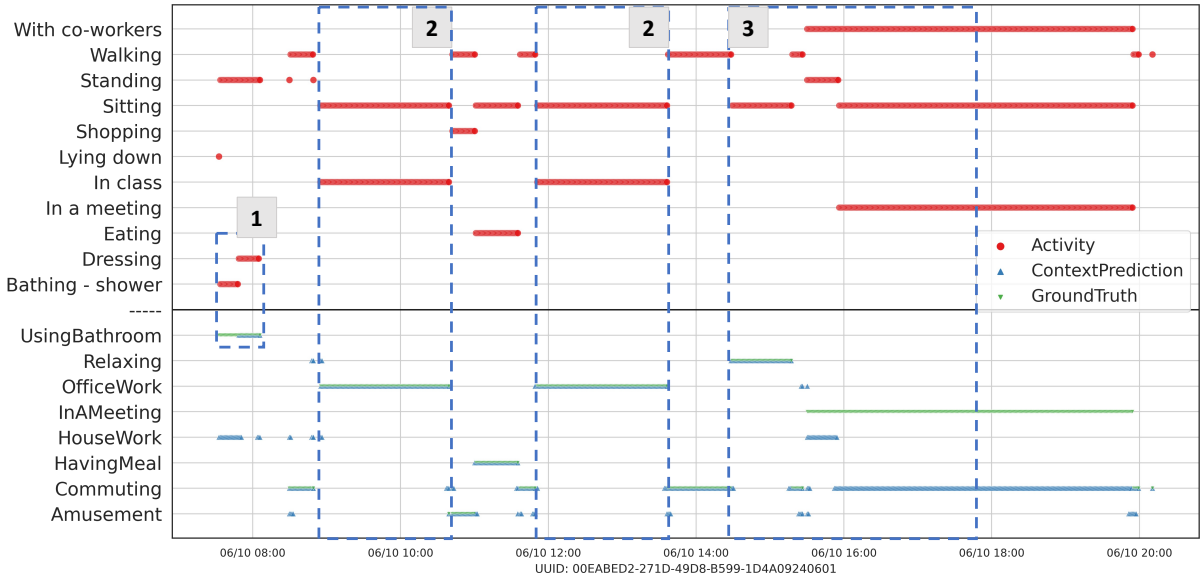


Fig. 18. A real-world timeline of different activities an individual performs from the Extrasensory datasets and the corresponding ground truth and context predictions from the ontology. The annotation in the graph shows different activity patterns, such as (a) sequential and (b) parallel activity patterns.

Figure 18 showcases real activity traces from the Extrasensory datasets showing different activity patterns, such as (a) sequential and (b) parallel activity patterns. Alongside the activity data, we also showcase the corresponding

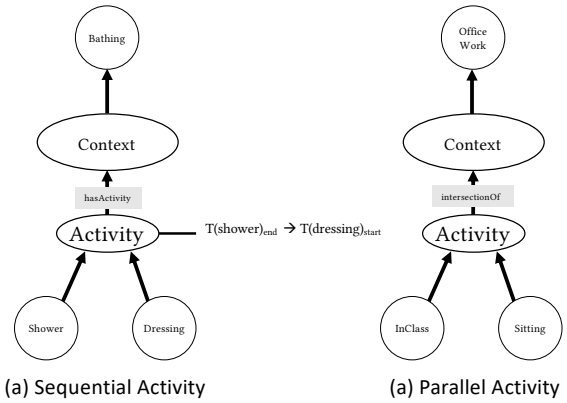


Fig. 19. Examples of sequential and parallel activity pattern definitions in the ontology

ground truth labels and context predictions derived from the ontology. This timeline graph provides insights into the patterns of activities and how they align with the expected context.

Ground-truth context annotation: While some public datasets have context annotations from sensor data, none of these datasets have an activity for context mapping. Hence, we created our own context labels using annotated activities from these datasets (from *Extrasensory* and *Casas*). We first normalized the vocabulary for the activity labels across the datasets using our ontology (e.g., *A: eat breakfast, A: eat lunch, A: eat dinner* were normalized to *A:'Eating'*). To bootstrap our context annotation process, we created timestamped activity traces as shown in Figure 18, which allowed us to visualize the patterns of activities and their correlation to an expected context (without context prediction label). Using the activity labels, we recognized 14 distinct contexts, and two researchers independently annotated 90 randomly sampled days of activity data across both datasets. Through this process, we not only achieved precise ground truth annotations but also discovered context patterns beyond the existing ontology, enabling us to expand and enhance our ontology.

Ontology prediction: In Figure 18, 1 shows an example of an activity pattern, where activity *A: Bathing-Shower* is followed by *A: Dressing*. This sequential transition has been explicitly modeled in the ontology of TAO, enabling us to infer the corresponding context *C: Using bathroom* (as depicted in Figure 19(a)) whenever this sequential activity sequence occurs. We also see the time difference between the ground truth annotation of the *C: Using bathroom* context and the actual model prediction. We see that the ontology model only predicts during the transition and not before that. We see similar cases in the other activity patterns and the corresponding activity prediction, as shown in 3 (*A: walking* to *A: sitting* –> *C: Commuting*, instead of *C: Meeting*). Similarly, number 2 shows a parallel activity pattern where A: Sitting is performed simultaneously with *A: inClass* activity. Our ontology correctly models these activities as *C: Office work* context because of the definition in the ontology (as depicted in Figure 19(b))

Overall, this analysis of real activity traces, along with the corresponding ground truth labels and context predictions derived from the ontology, allows us to gain valuable insights into the patterns and relationships between activities and their associated contexts, leading to a deeper understanding of the context-aware capabilities of our ontology.

A.2 Dataset Characteristics

In this section, we will examine the characteristics of the dataset used in our study. By exploring various aspects such as data size, sources, collection methods, and relevant statistical measures, we aim to provide a comprehensive overview of the dataset that TAO uses.

Extrasensory/CASAS:. Extrasensory dataset contains data from 60 participants, with thousands of samples per participant typically taken in intervals of 1 minute (but not necessarily in one long sequence, as there are time gaps). Every sample consists of sensor measurements from personal smartphones and self-reported labels for user activities and location. The *Casas* dataset is collected using ambient sensors deployed in 30 different smart homes and has 1 month of manually labeled activity information per home. We filtered activity annotations from these datasets to create multiple timestamped activity streams(at participant level in Extrasensory and home level in CASAS). We identified 17 different contexts from these activity labels. To evaluate the accuracy of our pipeline, we hand-annotated 85 days(selected randomly) of data across both datasets(more details in Table 4).

Table 4. Summary of the HAR datasets we use for our evaluation and their characteristics.

Dataset	Sensors	Total Users	Total Data Collected (in days)	Annotated Activities Data (in hours)	Total Unique Activities
ExtraSensory [67]	IMU,audio,location	60	471 days	86 hours	34
Casas [21]	Multimodal	30	2934 days	2850 hours	42

A.3 Detailed Scenarios for Real-world Study

In Table 5, we show details for 4 different scenarios we selected for our real-world study. All participants were asked to perform each of these scenarios and were not given any instruction on the duration of each activity, and we allowed them to complete the activity patterns at their own pace.

Table 5. Overview of the scenarios used in the real-world user study. We identify several common activity patterns in daily life from which context can be inferred and categorize them into different scenarios.

Scenario	Activity Patterns	Context
Scenario 1 (Sequential activity patterns)	*A: door knock* followed by *A: door open*	*C: ComingIn*
	A: phone ring followed by *A: talking*	*C: On a phone call*
	A: sweeping followed by *A: vacuum*	*C: House work*
Scenario 2 (Sequential and Parallel activity patterns)	*A: eating* followed by *A: typing*	*C: Having meal, C: Office work*
	A: coffee grinding, A: mouse click, A: typing	*C: Preparing meal, C: Office work*
	A: phone ring, A: talking, A: sweeping, A: coffee grinding	*C: Phone call, C: House work, C: Preparing meal*
Scenario 3 (Interleaved activity patterns for single context)	*A: jumping, A: running*	*C: Exercising*
	A: mouse click, A: typing, A: writing	*C: Office work*
	A: running, A: jumping	*C: Exercising*
	A: typing, A: mouse click, A: writing	*C: Office work*
Scenario 4 (Interleaved and parallel activity patterns for multiple contexts)	*A: tv, A: talking* and *A: typing*	*C: Relaxing, C: Office work*
	A: eating, A: mouse click, A: tv	*C: Having meal, C: Office work, C: Relaxing*
	A: vacuum, A: talking	*C: House work, C: Office work*

A.4 Details on Ground Truth Annotations

In table 6, we provide some set of example activities from the *Extrasensory* dataset and show corresponding manual annotation for ground truth.

Table 6. Example activities from the *Extrasensory* dataset [67]. We manually annotate the context for these activities during our ground truth labeling process.

Activity	Context
A: talking, A: computer work	*C: In a meeting, C: Office work*
A: jogging, A: cycling, A: running	*C: Exercise*
A: watching tv, A: talking	*C: Relaxing*

Table 7 presents a summary of our ground truth annotations, including 14 unique contexts annotated across *Extrasensory* and *Casas* datasets and the amount of data annotated for each context in terms of total time (in hours) and the number of unique instances (i.e., time intervals) for each context.

Table 7. Amount of data annotated for ground truth of 14 different contexts across *Extrasensory* and *Casas* datasets.

Context	Total Time(in hours)	Instance_Count
Amusement	82.9	45
ComingIn	0.44	88
Commuting	40.33	108
Exercising	6.27	20
GoingOut	1.33	107
HavingMeal	58.52	218
HouseWork	27.94	89
InAMeeting	48.97	55
OfficeWork	96.39	112
PhoneCall	3.30	21
PreparingMeal	20.27	87
Relaxing	258.69	265
Sleeping	561.52	203
UsingBathroom	82.43	382