Chapter 1

ENHANCING ANOMALY DETECTION USING TEMPORAL PATTERN DISCOVERY

Vikramaditya R. Jakkula
Washington State University
Box 642752, Pullman, Washington, 99164-2752, USA
vjakkula@eecs.wsu.edu

Aaron S. Crandall
Washington State University
Box 642752, Pullman, Washington, 99164-2752, USA
acrandal@eecs.wsu.edu

Diane J. Cook
Washington State University
Box 642752, Pullman, Washington, 99164-2752, USA
cook@eecs.wsu.edu

Abstract Technological enhancements aid development and research in smart homes and intelligent environments. The temporal nature of data collected in a smart environment provides us with a better understanding of patterns that occur over time. Predicting events and detecting anomalies in such datasets is a complex and challenging task. To solve this problem, we suggest a solution using temporal relations. Our temporal pattern discovery algorithm, based on Allen's temporal relations, has helped discover interesting patterns and relations from smart home datasets. We hypothesize that machine learning algorithms can be designed to automatically learn models of resident behavior in a smart home, and when these are incorporated with temporal information, the results can be used to detect anomalies. We describe a method of discovering temporal relations in smart home datasets and applying them to perform anomaly detection on the frequently-occurring events by incorporating temporal relation information shared by the activity. We validate our hypothesis using empirical studies based on the data collected from real resident and virtual resident (synthetic) data.

Keywords: anomaly detection, temporal relationships, smart environments

Introduction

The problems of representing, discovering, and using temporal knowledge arise in a wide range of disciplines, including computer science, philosophy, psychology, and linguistics. Temporal rule mining and pattern discovery applied to time series data has attracted considerable interest over the last few years. We consider the problem of learning temporal relations between event time intervals in smart environment data, which includes physical activities (such as taking pills while at home) and instrumental activities (such as turning on lamps and electronic devices). These learned temporal relations can be used to detect anomalies. The purpose of this work is to identify interesting temporal patterns in order to detect whether the event which occurred is an anomaly. A simple sensor can produce an enormous amount of temporal information, which is difficult to analyze without temporal data mining techniques that are developed for this purpose.

Our vision is to keep older adults functioning independently in their own homes longer. The number of Americans who live with cognitive or physical impairments is rising significantly due to the aging of the population and better medical care. By 2040, an estimated 23% of the US population will be 65+ (Lanspery et al., 1997). Many of these elder adults live in rural areas with limited access to health care. While 90% of Americans over 60 want to live out their lives in familiar surroundings (Gross, 2007), today those who need special care often must leave home to meet medical needs. Providing this care at home will become a requirement because 40% of elder adults cannot afford to live in assisted care facilities, and because hospitals and nursing homes do not have the capacity to handle the coming “age wave” of a larger, sicker population (Wang, 2006).

Data collected in smart environments has a natural temporal component to it, and reasoning about such timing information is essential for performing tasks such as anomaly detection. Usually, these events can be characterized temporally and can be represented by time intervals. These temporal units can also be represented using their start time and end time which lead to form a time interval, for instance when the cooker is turned on it can be referred to as the start time of the cooker and when the cooker is turned off it can be referred to as the end time of the cooker. The ability to provide and represent temporal information at different levels of granularity is an important research sub-field in computer science which especially deals with large timestamp datasets. The representation and reasoning about temporal knowledge is very essential for smart home applications. Individuals with disabilities, elder adults and chronically ill individuals can take advantage of applications that use temporal knowledge. In particular, we can model activities of these individuals, use this information to distinguish normal activities from abnormal activities and help make critical decisions to ensure their safety.

The objective of this research is to identify temporal relations among daily activities in a smart home to enhance prediction and decision making with these discovered relations, and detect anomalies. We hypothesize that machine learning algorithms can be designed to automatically learn models of resident behavior in a smart home, and when these are incorporated with temporal information, the results can be used to detect anomalies. We discuss Allen's notion of temporal relationships and describe how we can discover frequently-occurring temporal relationships in smart home data. We then use the discovered temporal relations to perform anomaly detection. We validate our hypothesis using empirical studies based on the data collected from real resident and synthetic data.

1. Temporal Reasoning

Activities in a smart home include resident activities as well as interactions with the environment. These may include walking, sitting on a couch, turning on a lamp, using the coffee maker, and so forth. Instrumental activities are those which have some interaction with an instrument which is present and used in the environment. We see that these activities are not instantaneous, but have distinct start and end times. We also see that there are well-defined relationships between the time intervals for different activities. These temporal relations can be represented using Allen's temporal relations and can be used for knowledge and pattern discovery in day-to-day activities. These discoveries can be used for developing systems which detect anomalies and aid caregivers in taking preventive measures.

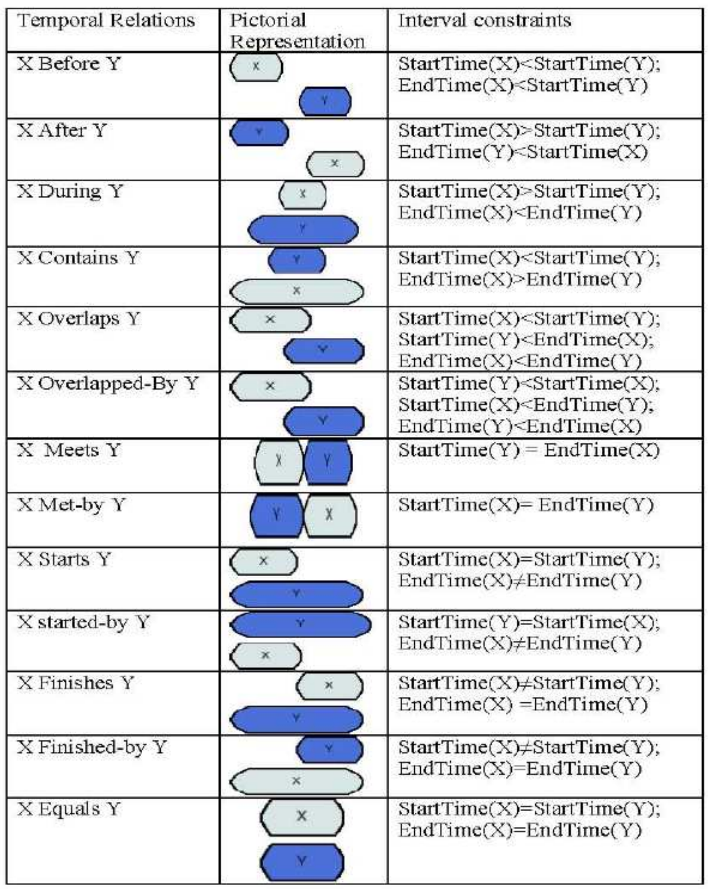
Allen (Allen and Ferguson, 1994) listed thirteen relations (visualized in Figure 1.1) comprising a temporal logic: before, after, meets, meet-by, overlaps, overlapped-by, starts, started-by, finishes, finished-by, during, contains, and equals. These temporal relations play a major role in identifying time-sensitive activities which occur in a smart home. Consider, for example, a case where the resident turns the television on before sitting on the couch. We notice that these two activities, turning on the TV and sitting on the couch, are frequently related in time according to the "before" temporal relation. Modeling temporal events in smart homes is an important problem and offers benefits to residents of smart homes. Temporal constraints can be useful when reasoning about activities; if a temporal constraint is not satisfied then a potential "anomalous" or "critical" situation may have occurred.

Temporal mining is a relatively new area of research in computer science and has become more popular in the last decade due to the increased ability of computers to store and process large datasets of complex data. Temporal reasoning and data mining has been investigated in the context of classical and temporal logics and have been applied to real-time artificial intelligence systems.

Morchen argued that Allen's temporal patterns are not robust and small differences in boundaries lead to different patterns for similar situations (Morchen, 2006). Morchen presents a Time Series Knowledge Representation (TSKR), which expresses the temporal concepts of coincidence and partial order. He states that Allen's temporal relations are ambiguous in nature, making them not scalable and not robust. Morchen handles this problem of ambiguity by applying constraints to define the temporal relations. Although this method appears feasible, it does not suit our smart home application due to the granularity of the time intervals in smart homes datasets. In smart environments, some events are instantaneous while others span a long time period. Morchen applies TSKR to muscle reflection motion and other applications where time intervals are consistently similar in length. Because the TSKR approach also does not eliminate noise and is computationally expensive, the approach is not well suited to the large and complex sensor data that is created by smart environments.

In artificial intelligence, the event calculus is a frequently-used approach for representing and reasoning about events and their effects. Gottfried, et al. (Gottfried et al., 2006) also argue that space and time play essential roles in everyday lives and introduce time and space calculi to reason about these dimensions. They discuss several AI techniques for dealing with temporal and spatial knowledge in smart homes, mainly focusing on qualitative approaches to spatiotemporal reasoning.

Ryabov and Puuronen (Ryabov and Puuronen, 2001) in their work on probabilistic reasoning about uncertain relations between temporal points represent the uncertain relation between two points by an uncertainty vector with three probabilities of basic relations ("$<$", "$+$", and "$>$"). They also incorporate inversion, composition, addition, and negation operations into their reasoning mechanism. This model would



Temporal Relations	Pictorial Representation	Interval constraints
X Before Y		StartTime(X)<StartTime(Y); EndTime(X)<StartTime(Y)
X After Y		StartTime(X)>StartTime(Y); EndTime(Y)<StartTime(X)
X During Y		StartTime(X)>StartTime(Y); EndTime(X)<EndTime(Y)
X Contains Y		StartTime(X)<StartTime(Y); EndTime(X)>EndTime(Y)
X Overlaps Y		StartTime(X)<StartTime(Y); StartTime(Y)<EndTime(X); EndTime(X)<EndTime(Y)
X Overlapped-By Y		StartTime(Y)<StartTime(X); StartTime(X)<EndTime(Y); EndTime(Y)<EndTime(X)
X Meets Y		StartTime(Y) = EndTime(X)
X Met-by Y		StartTime(X)= EndTime(Y)
X Starts Y		StartTime(X)=StartTime(Y); EndTime(X)≠EndTime(Y)
X started-by Y		StartTime(Y)=StartTime(X); EndTime(X)≠EndTime(Y)
X Finishes Y		StartTime(X)≠StartTime(Y); EndTime(X) =EndTime(Y)
X Finished-by Y		StartTime(X)≠StartTime(Y); EndTime(X)=EndTime(Y)
X Equals Y		StartTime(X)=StartTime(Y); EndTime(X)=EndTime(Y)

Figure 1.1. Thirteen temporal relationships comprising Allen's temporal logic.

not be suitable for a smart home scenario as it would not delve into finer granularities to analyze instantaneous events. The work of Worboys and Duckham (Worboys and Duckham, 2002) involves spatio-temporal-based probability models, the implementation of which is currently identified as future work. Dekhtyar, et. al.'s research on probabilistic temporal databases (Dekhtyar et al., 2001) provides a framework which is an extension of a relational algebra that integrates both probabilities and time. This work, like ours, builds on Allen's temporal logic.

2. The MavHome Smart Home

Our anomaly detection algorithm is designed as part of the MavHome smart home project (Youngblood and Cook, 2007; Youngblood et al., 2005). We view a smart environment as an intelligent agent, which determines the state of the environment using sensors and acts upon the environment using powerline controllers. All of the MavHome components are implemented and have been tested in two physical environments, the MavLab workplace environment and an on-campus apartment. Powerline control automates all lights and appliances, as well as HVAC, fans, and miniblinds. Perception of light, humidity, temperature, smoke, gas, motion, and switch settings is performed through a sensor network developed in-house. Inhabitant localization is performed using passive infrared sensors yielding a detection rate of 95% accuracy.

The MavHome architecture shown in Figure 1.2 consists of cooperating layers. Perception is a bottom-up process. Sensors monitor the environment using physical components (e.g., sensors) and make information available through the interface layers. The database stores this information while other information components process the raw information into more useful knowledge (e.g., patterns, predictions). New information is presented to the decision making applications (top layer) upon request or by prior arrangement. Action execution flows top-down. The decision action is communicated to the services layer which records the action and communicates it to the physical components. The physical layer performs the action using powerline control, and other automated hardware, thus changing the state of the world and triggering a new perception.

Communication between high-level components is performed using CORBA, and each component registers its presence using zero configuration (ZeroConf) technologies. Implemented services include a PostgreSQL database that stores sensor readings, prediction components, data mining components, and logical proxy aggregators. Resource uti-

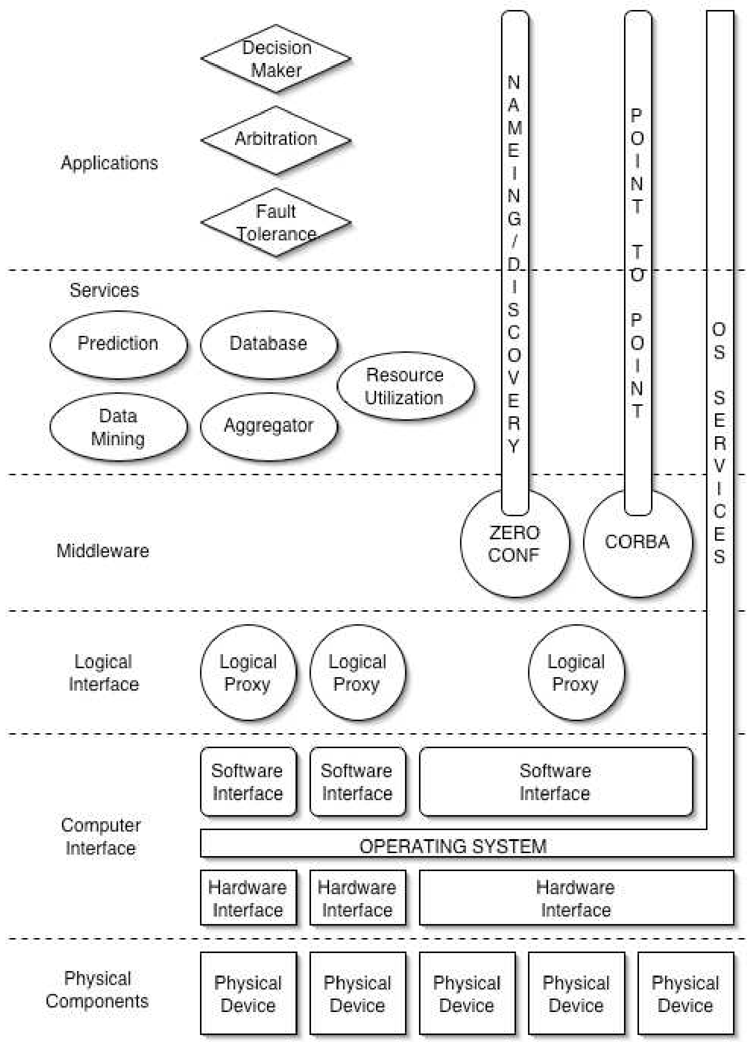


Figure 1.2. MavHome architecture.

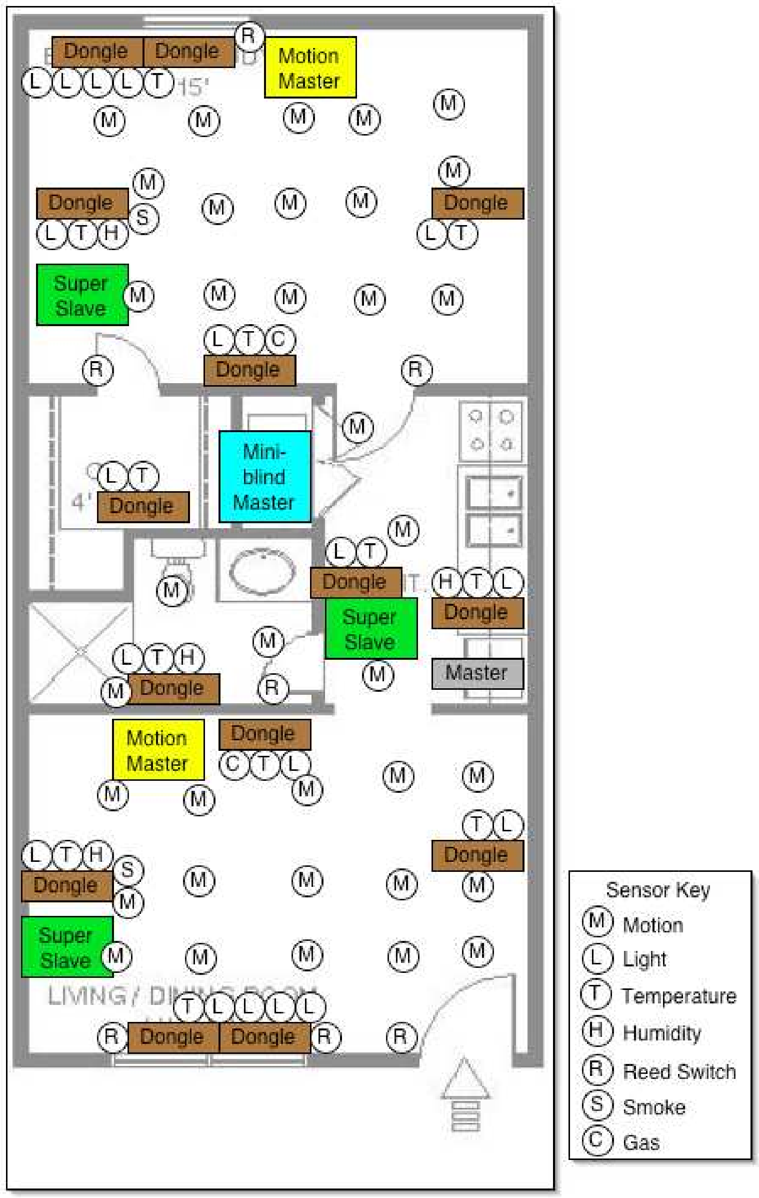


Figure 1.3. MavPad sensor layout.

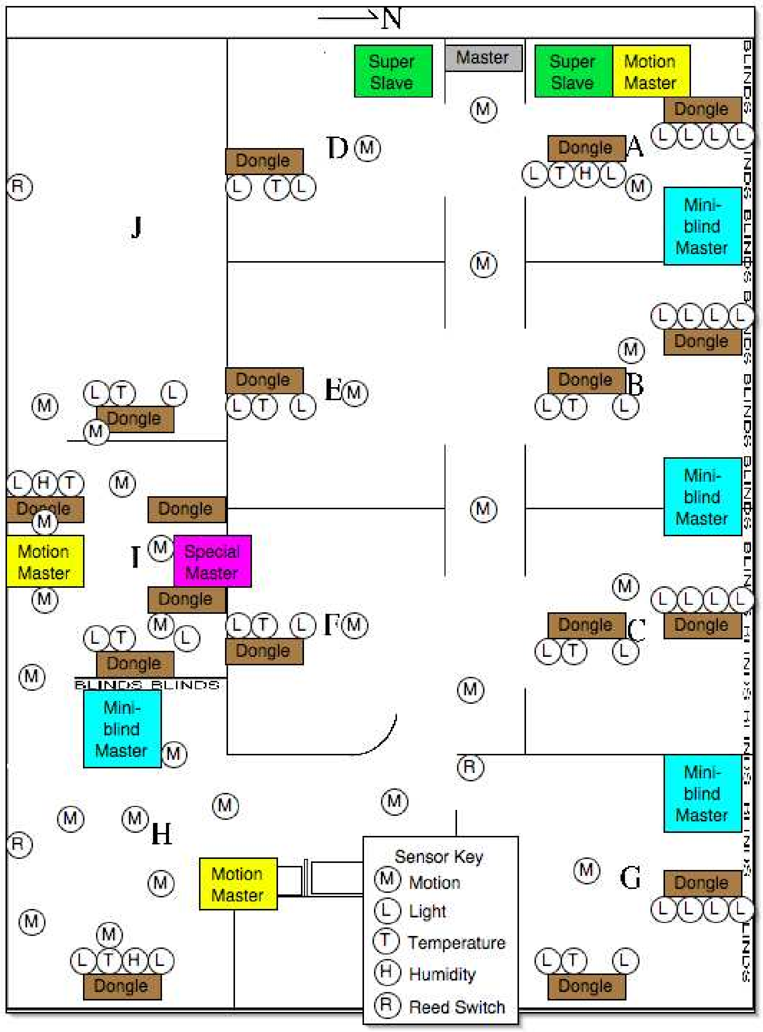


Figure 1.4. MavLab sensor layout.

lization services monitor current utility consumption rates and provide usage estimates and consumption queries.

MavHome is designed to optimize a number of alternative functions, but we initially focused on minimization of manual interactions with devices. The MavHome components are fully implemented and were used to automate the environments shown in Figure 1.3 through 1.5. The MavLab environment contains work areas, cubicles, a break area, a lounge, and a conference room. MavLab is automated using 54 X-10 controllers and the current state is determined using light, temperature, humidity, motion, and door/seat status sensors. The MavPad is an on-campus apartment hosting a full-time student occupant. MavPad is automated using 25 controllers and provides sensing for light, temperature, humidity, leak detection, vent position, smoke detection, CO detection, motion, and door/window/seat status sensors.

To automate our smart environment, we collect observations of manual inhabitant activities and interactions with the environment. We then mine sequential patterns from this data using a sequence mining algorithm, ED. Next, our ALZ algorithm predicts the inhabitant's upcoming actions using observed historical data. Finally, a hierarchical Markov model is created by our ProPHeT algorithm using low-level state information and high-level sequential patterns, and is used to learn an action policy for the environment. Figure 1.6 shows how these components work together to improve the overall performance of the smart environment. In our initial study, we were able to use these software components and data collected in the smart environments to identify frequent resident behavior patters, predict sensor events, and ultimately automate 76% of the resident's interactions with the environments (Youngblood and Cook, 2007).

Our initial MavHome implementation and experiments indicated that it is possible to analyze and predict resident activities and to use this information for environment automation. This technology finds application in resident health monitoring as well. For this application, however, we see that the software algorithms could be improved by making use of timing information and temporal relationships to improve event prediction and to perform anomaly detection. Both of these features will allow MavHome to do a more effective job of monitoring the safety of the environment and its residents. In the next section we introduce a suite of software tools that is designed to provide these needed features.

3. TempAl

TempAl (pronounced as "temple") is a suite of software tools which enrich smart environment applications by incorporating temporal relationship information for various applications including anomaly detection. In smart homes, the time when an event takes place is known and is recorded. Our earlier MavHome algorithms did not incorporate time into its data analysis. We hypothesize that including this information would improve the performance of the smart home algorithms, which motivates our contributions of storing, representing, and analyzing timing information. The temporal nature of the data provides us with a better

Figure 1.5. MavLab rooms.

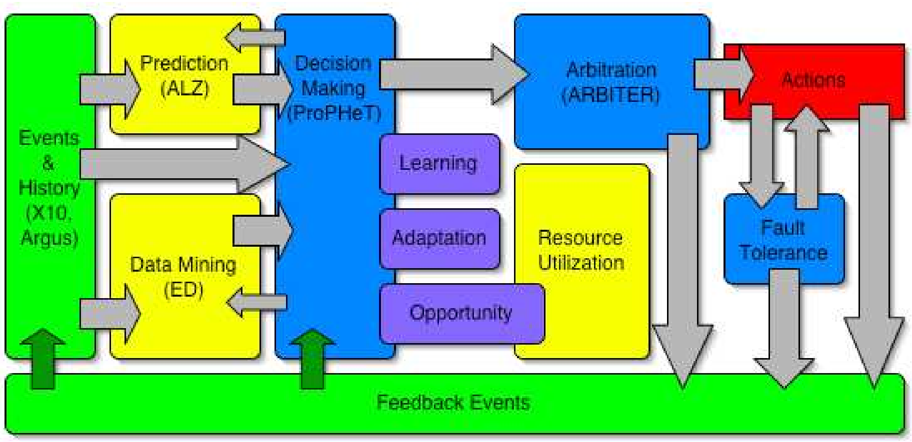


Figure 1.6. Integration of AI techniques into MavHome architecture.

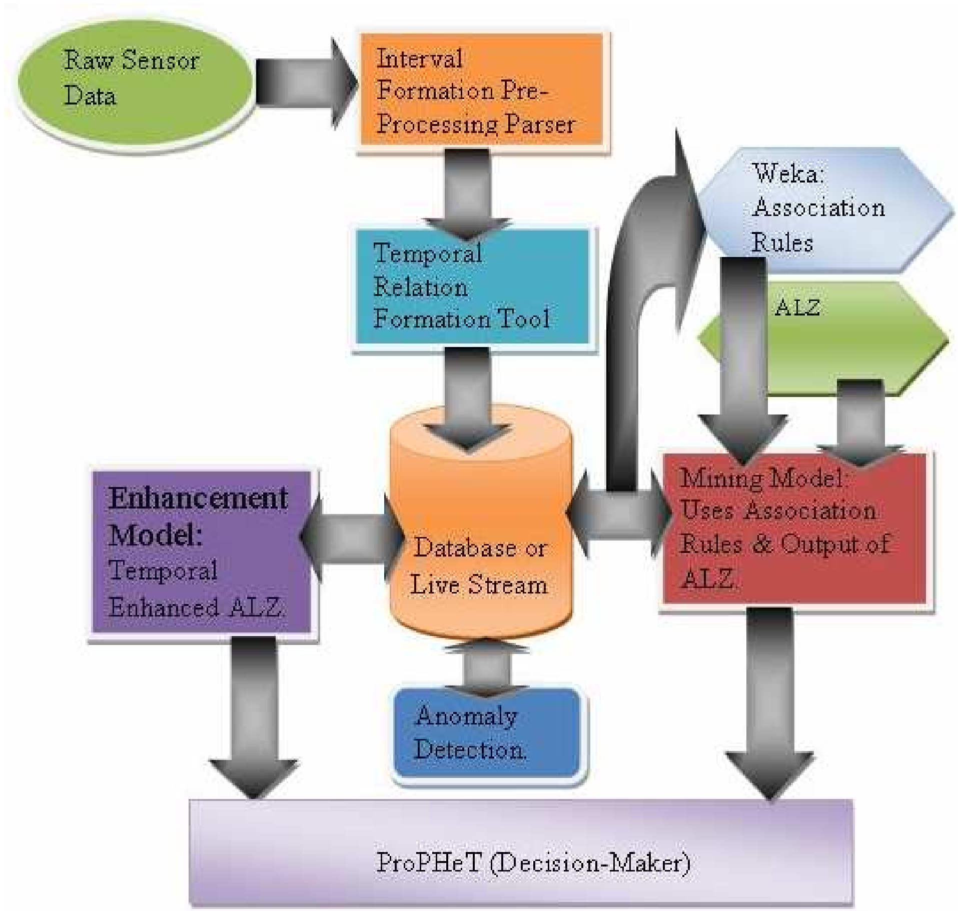


Figure 1.7. Architecture overview of TempAl.

understanding of the nature of the data. We see that using a time series model is a common approach to reasoning about individual time-based events. However, we consider events and activities using time intervals rather than time points, which is appropriate for smart environment scenarios. As a result, we have developed methods for finding interesting temporal patterns as well as for performing anomaly detection based on these patterns.

The architecture of TempAl and its integration with the MavCore is shown in Figure 1.7. Raw data is read and processed by a parser to identify interval data, which is later read by a temporal relations formulation tool to identify the temporal relations. The temporal relations data is later used by the anomaly detection and event prediction components, to enhance the performance of these individual algorithms.

The objective of this study is to determine if anomalies can be effectively detected in smart home data using temporal data mining. Specifically, we introduce a temporal representation that can express frequently-occurring relationships between smart environment events. We then use the observed history of events to determine the probability that a particular event should or should not occur on a given day, and report as an anomaly the presence (or absence) of highly-unlikely (highly likely) events.

The need for a robust anomaly detection model is as essential as a prediction model for any intelligent smart home to function in a dynamic world. For a smart environment to perform anomaly detection, it should be capable of applying the limited experience of environmental event history to a rapidly changing environment, where event occurrences are related by temporal relations. For example, if we are monitoring the well being of an individual in a smart home and the individual has not opened the refrigerator after they get out of bed as they normally do, this should be reported to the individual and the caregiver. Similarly, if the resident turned on the bathwater, but has not turned it off before going to bed, the resident or the caregiver should be notified, and the smart home could possibly intervene by turning off the water.

Identification of frequent temporal relationships

Anomaly detection is most accurate when it is based on behaviors that are frequent and predictable. As a result, we look for temporal interactions only among the most frequent activities that are observed in resident behavior. This filtering step also greatly reduces the computational cost of the algorithm. To accomplish this task, we mine the data for frequent sequential patterns using a sequence mining version of the Apriori algorithm (Agrawal and Srikant, 1995). The input to the algorithm is a file of sensor events, each tagged with a date and time, and the result is a list of frequently-occurring events, which occur most frequently among the inputted file of sensor events. Pseudocode for the algorithm is given below.

C_k: Candidate itemset of size k
L_k: Frequent itemset of size k
L_1: {frequent items};
For (k=1; $L_k \neq \emptyset$; k++)
do

C_{k+1} = candidates generated from L_k;
For each day t in dataset
do
Increment the count of all candidates in C_{k+1} that are contained in t
end
L_{k+1} = candidates in C_{k+1} with min_support
end
Return $\bigcup_k L_k$;

Next, we identify temporal relations that occur between events in these frequent sequences. The final step involves calculating the probability of a given event occurring (or not occurring), which forms the basis for anomaly detection.

Detecting anomalies

The temporal relations that are useful for anomaly detection are the before, contains, overlaps, meets, starts, started-by, finishes, finished-by, and equals relations shown in Figure 1.1. Because we want to detect an anomaly as it occurs (and not after the fact), the remaining temporal relations - after, during, overlapped-by, and met-by - are not included in our anomaly detection process.

Let us focus now on how to calculate the probability that event C will occur (in this case, the start of the event interval). Evidence for this probability is based on the occurrence of other events that have a temporal relationship with C, and is accumulated over all such related events. First consider the probability of C occurring given that the start of the temporal interval for event B has been detected. The formula to calculate the probability of event C based on the occurrence of event B and its temporal relationship with C is given by the equation:

$$P(C|B) = (|Before(B,C)| + |Contains(B,C)| + |Overlaps(B,C)|+$$
$$|Meets(B,C)| + |Starts(B,C)| + |StartedBy(B,C)|+$$
$$|Finishes(B,C)| + |FinishedBy(B,C)| + |Equals(B,C)|)/|B|$$

Note that the probability of B is based on the observed frequency of the observed temporal relationships between B and C as well as the number of occurrences of B in the collected event history. In this equation, we compute the probability of the occurrence of C given that B occurred using the temporal relations frequency shared between the two events. This probability count includes only those relations which aid anomaly detection. These values are added as they do not overlap and the constraints strictly enforces bounds to check that the relations are unique and thus the probability count includes the sum of their occurrences.

The previous discussion showed how to calculate the likelihood of event C given the occurrence of one other event B. Now consider the case where we want to combine evidence from multiple events that have a temporal relationship with C. In our example we have observed the start of event A and the start of event B, and want to establish the likelihood of event C occurring. The combined probability is computed as:

$$P(C|A \cup B) = P(C \cap (A \cup B))/P(A \cup B)$$

In this equation we calculate the probability of C occurring (here C is the most recently occurred event) when A and B are both frequent events and both have occurred. When we find the occurrences of A and B they might have some temporal relations in common (i.e., a that A has with B and an inverse relationship that B has with A). In these cases, one of the relationships is removed to avoid repetitive counts. An alternative computation would be $P(AB|C)$, which would be interpreted as given that C occurred, determine whether A and B are anomalies. Our approach looks at the most current observed event C, and calculates evidence supporting the claim that C is an anomaly.

Such anomaly detection is useful in health monitoring. When combined with a decision maker, a smart environment could respond to a detected anomaly by reminding the resident of the needed event or automating the event. Using this second equation we can calculate the likelihood of event C occurring based on every event we have observed on a given day to that point in time. We also need to note that for the anomaly detection process we consider that each day starts with a blank slate and as the events occur new anomaly values are computed. We can similarly calculate the likelihood that an event C would not occur as $P(\neg C) = 1 - P(C)$. Finally, we calculate the anomaly value of event C using the equation $Anomaly_C = 1 - P(C)$.

Note that if the event has a anomaly probability approaching 1 and the event occurred, this is considered an anomaly. Similarly, if the probability is close to 0 and the event does not occur then it should also be considered an anomaly. The point at which these anomalies are considered surprising enough to be reported is based somewhat on the data itself (Noble and Cook, 2003). If the probability of an event is based on the occurrence of other events which themselves rarely occur, then the evidence supporting the occurrence of the event is not as strong. In this case, if the event has a low probability yet does occur, it should be considered less anomalous than if the supporting evidence itself appears with great frequency. Consistent with this theory, we calculate the mean and standard deviation of event frequencies over the set of frequent events in the resident's action history. An event (or, conversely, the absence of an event) are reported as an anomaly if it does (does not) occur and its anomaly value is greater (smaller) than the mean probability + 2 standard deviations (or mean - 2 standard deviations). Two standard deviations away from the mean accounts for roughly 95%, so any value which falls out of this population would be reported as an anomaly.

To illustrate the process, we start with a sample of raw data collected in the MavLab environment:

Timestamp	Sensor State	Sensor ID
...		
3/3/2003 11:18:00 AM	OFF	E16
3/3/2003 11:23:00 AM	ON	G12
3/3/2003 11:23:00 AM	ON	G11
3/3/2003 11:24:00 AM	OFF	G12
3/3/2003 11:24:00 AM	OFF	G11
3/3/2003 11:24:00 AM	ON	G13
3/3/2003 11:33:00 AM	ON	E16
3/3/2003 11:34:00 AM	ON	D16
3/3/2003 11:34:00 AM	OFF	E16
...		

Next, we identify (start, end) time intervals that are associated with the events. Here is a sample of the time intervals that are associated with the raw data:

Date	Sensor ID	Start Time	End Time
03/02/2003	G11	01:44:00	01:48:00
03/02/2003	G13	04:06:00	01:48:00
03/03/2003	E16	11:18:00	11:34:00
03/03/2003	G12	11:23:00	11:24:00

Once the time intervals are established we discover temporal relations that frequently exist among these events, such as the ones shown here:

Datasets	# Days	# Events	# Identified Intervals	Dataset Size
Real	60	17	1623	104KB
Synthetic	60	8	1729	106KB

Table 1.1. Parameter settings for experiments.

Time	Sensor ID	Temporal Relation	Sensor ID
3/3/2003 12:00:00 AM	G12	DURING	E16
3/3/2003 12:00:00 AM	E16	BEFORE	I14
3/2/2003 12:00:00 AM	G11	FINISHESBY	G11
4/2/2003 12:00:00 AM	J10	STARTSBY	J12

The frequencies of these relationships are tabulated and used as the basis for calculating anomaly values each time a new event is observed. When an event occurs which has a sufficiently high anomaly value, the event is reported as an anomaly.

4. Experimental Findings

To validate our TempAl anomaly detection algorithm, we apply it to real and synthetic smart home data. Table 4 summarizes features of the datasets used for these experiments. The real data represents raw sensor data collected for 60 days in the MavLab environment with a volunteer resident. The synthetic data represents instances of a predefined set of activities. In the synthetic data we have injected anomalous events to see if TempAl will catch these events and label them as anomalies.

To test our algorithms we train the models using 59 days of sensor data, then test the model on a single day of events. Table 1.1 shows the anomaly values that are calculated for the eight observed events in the real data, and Table 1.2 shows the anomaly values for the seventeen observed events in the synthetic data. The values are visualized in Figure 1.8.

Based upon a visual inspection of the data we see that the anomaly detection algorithm performed well — all of the expected anomalies were detected and no false positives were reported. In the real data no anomalies are reported, which is consistent with the nature of the data. This result reflects the fact that anomalies should be, and are in fact, rare. We see that the TempAl algorithms are robust in this case and do not report false positives.

The experimental results summarized here provide evidence that our algorithm is capable of identifying anomalous events based on temporal

Frequent Event ID (in chronological order)	Frequent Event	Event Probability	Anomaly Value	Anomaly Detected
1	J10	0.45	0.55	No
2	J11	0.32	0.68	No
3	A11	0.33	0.67	No
4	A15	0.24	0.76	No
5	A11	0.23	0.77	No
6	A15	0.22	0.78	No
7	I11	0.27	0.73	No
8	I14	0.34	0.66	No
Anomaly Mean			0.7	
Anomaly Standard Deviation			0.07	
Anomaly Threshold			0.84	

Table 1.2. Anomaly detection in real data test set.

Frequent Event ID (in chronological order)	Frequent Event	Event Probability	Anomaly Value	Anomaly Detected
1	Lamp	0.30	0.70	No
2	Lamp	0.23	0.77	No
3	Lamp	0.01	0.99	Yes
4	Fan	0.32	0.68	No
5	Cooker	0.29	0.71	No
6	Lamp	0.45	0.55	No
7	Lamp	0.23	0.77	No
8	Lamp	0.01	0.99	Yes
9	Lamp	0.23	0.77	No
10	Fan	0.30	0.70	No
11	Cooker	0.34	0.66	No
12	Lamp	0.33	0.67	No
13	Lamp	0.20	0.80	No
14	Lamp	0.02	0.98	No
15	Lamp	0.00	1.00	Yes
16	Fan	0.34	0.66	No
17	Cooker	0.42	0.58	No
Anomaly Mean			0.76	
Anomaly Standard Deviation			0.14	
Anomaly Threshold			0.99	

Table 1.3. Anomaly detection in synthetic data test set.

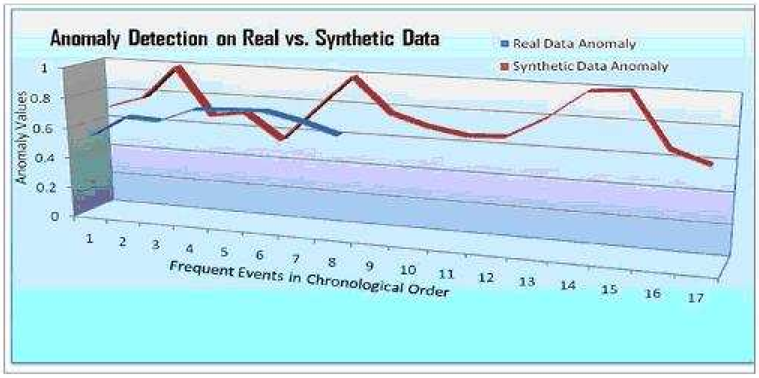


Figure 1.8. Visualization of calculated anomaly values for real and synthetic data. The spikes in the graph are events which are flagged as anomalies.

relationship information. The results applied to real data bring insights to the activities that were being performed in the MavLab setting. In both cases the anomalies would be reported to the resident and possibly their caregiver. The caregiver could respond according to the health-critical nature of the anomaly and any additional information they possess.

An extended application of anomaly detection is its use for reminder assistance. If the resident queries the algorithm for the next routine activity, the expected activity or activities with the greatest probability can be provided. Similarly, if an anomaly is detected, the smart environment can first initiate contact with the resident and provide a reminder of the activity that is usually performed at that time. Autominder (Pollack et al., 2003) is an example of a reminder system that has already been developed for this purpose using techniques such as dynamic programming and Bayesian learning. Unlike our approach, Autominder does not base its generated reminders on a model of behavior that is learned from actual observed events.

5. Conclusions

Temporal reasoning enhances smart environments algorithms by incorporating learned information about temporal relationships between events in the environment. Based on our study, we conclude that the use of temporal relations provides us with an effective new approach for anomaly detection. We tested TempAl on relatively small datasets, but will next target larger datasets with real data collected over a six month time span.

Another major enhancement to this work would be to consider an interval analysis of intermediate device states. Intermediate states are those which exist between an ON and OFF state. For example, a lamp controlled by a dimmer switch will have a number of light levels supported by the dimmer, which form the intermediate states. Identifying time intervals for events changing intermediate states would be a challenge for TempAl but would provide more refined information to the algorithms.

In addition, we would like to investigate other techniques for identifying the anomaly threshold function. Other future work can focus on finding better fusion techniques to enhance existing anomaly detection algorithms using temporal relationship information.

While making sense of sensor data can be challenging for smart environment algorithms, the problem is made even more complex when the environment houses more than one resident (Jakkula et al., 2007). To aid the capabilities of our temporal data mining algorithms, and to reveal the complexities of multi-resident spaces, an entity discovery tool is needed. Enriching the raw data set provided by the smart environment gives the knowledge discovery tools more information to use during the mining process. This comes in the form of an entity (in this case, resident) identification number that is attached to each event, matching events to entities. Thus, using temporal activity models to identify patterns and associate these patterns to behavior models for entity identification and resident profiling is a direction we are currently pursuing.

Agents in dynamic environments have to deal with changes over time. Enhancing TempAl to detect changes in temporal relationship frequencies and to use this information would be a good future direction of this work.

Acknowledgments

This work was partially supported by National Science Foundation grant IIS-0121297.

References

Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In *Proceedings of the International Conference on Data Engineering*, pages 3–14.

Allen, J. F. and Ferguson, G. (1994). Actions and events in interval temporal logic. *Journal of Logic and Computation*, 4(5):531–579.

Dekhtyar, A., Ross, R., and Subrahmanian, V. S. (2001). Probabilistic temporal databases, i: Algebra. *ACM Transactions on Database Systems*, 26(1):41–95.

Gottfried, B., Guesgen, H. W., and Hubner, S. (2006). Spatiotemporal reasoning for smart homes. In *Designing Smart Homes*, pages 16–34. Springer Berlin / Heidelberg.

Gross, J. (August 14, 2007). A grass-roots effort to grow old at home. In *The New York Times*.

Jakkula, V., Crandall, A., and Cook, D. J. (2007). Knowledge discovery in entity based smart environment resident data using temporal relations based data mining. In *Proceedings of the ICDM Workshop on Spatial and Spatio-Temporal Data Mining*.

Lanspery, S., Callahan, Jr., J. J., Miller, J. R., and Hyde, J. (1997). Introduction: Staying put. In *Staying Put: Adapting the Places Instead of the People*, pages 1–22. Baywood Publishing Company.

Morchen, F. (2006). Algorithms for time series mining. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 668–673.

Noble, C. and Cook, D. J. (2003). Graph-based anomaly detection. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Pollack, M., Brown, L., Colbry, D., McCarthy, C., Orosz, C., Peintner, B., Ramakrishnan, S., and Tsamardinos, I. (2003). Autominder: An intelligent cognitive orthotic system for people with memory impairment. *Robotics and Autonomous Systems*, 44(3-4):273–282.

Ryabov, V. and Puuronen, S. (2001). Probabilistic reasoning about uncertain relations between temporal points. In *Proceedings of the Eighth*

International Symposium on Temporal Representation and Reasoning, pages 35–40.

Wang, H. (2006). Digital home health - a primer. In *Parks Associates*.

Worboys, M. F. and Duckham, M. (2002). Integrating spatio-thematic information. In *Proceedings of the International Confernece on Geographic Information Science*, pages 346–361.

Youngblood, G. M. and Cook, D. J. (2007). Data mining for hierarchical model creation. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 37(4):1–12.

Youngblood, G. M., Holder, L. B., and Cook, D. J. (2005). Managing adaptive versatile environments. *Journal of Pervasive and Mobile Computing*, 1(4):373–403.