ENHANCING SMART HOME RESIDENT ACTIVITY PREDICTION AND ANOMALY DETECTION USING TEMPORAL RELATIONS

By

VIKRAMADITYA REDDY JAKKULA

A thesis submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE in COMPUTER SCIENCE

WASHINGTON STATE UNIVERSITY
Department of Electrical Engineering and Computer Science

DECEMBER 2007

To the Faculty of Washington State University:

The members of the Committee appointed to examine the thesis of VIKRAMADITYA REDDY JAKKULA find it satisfactory and recommend that it be accepted.

Chair

ACKNOWLEDGMENT

I would like to thank my advisor and mentor Dr. Diane J. Cook, who along with Dr. Lawrence B. Holder stands as a constant source of inspiration to me. I enjoyed working on challenging areas of intelligent environment research under Dr. Cook's supervision. I still remember the initial email communication which I did have with Dr. Cook before I started my graduate studies, which were filled with encouragement and a promise of guidance. And I would always remember her kindness of inviting me to lunch and sharing research ideas and ways to approach research problems. I would also like to thank Dr. Holder for his advice, and all the conversations with Dr. Holder were cheerful and encouraging. I would also like to thank Dr. Michael Youngblood for his inspiration, guidance and advices, which I received during my initial year of graduate study. I would always remember him as the person who taught me to question the how, what and why of any research problem that I face.

I would also like to thank my faculty at Washington State University which includes Dr. Sirisha Medidi, and other computer science faculty with whom I had numerous amounts of discussion on new trends in research and technology. I would also like to thank all my colleagues and fellow researchers which include Mr. Gaurav Jain, Mr. Darin Breazle, Mr. Nikhil Ketkar, Mr. Chang Hun, Dr. Mehran Asadi, Mr. Aaron Crandall, Mr. Denis Gjoni and all others who were a part of the University of Texas, Arlington and Washington State University AI Labs, for their interactions and discussions, as I believe they were a constant source of feedback for my progressing research. Thank you for your kind remarks! This work was supported by NSF grant IIS-0121297. I would also like to thank NETRONIC Software GmbH for providing me a temporary license to their developer API for the visualizer tool.

ENHANCING SMART HOME RESIDENT ACTIVITY PREDICTION AND ANOMALY DETECTION USING TEMPORAL RELATIONS

Abstract

by Vikramaditya Reddy Jakkula, M.S.
Washington State University
December 2007

Chair: Diane J. Cook.

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 on 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 enhance prediction and 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 and enhance sequential prediction 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 (or synthetic) data.

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENTS iii

ABSTRACT iv

LIST OF TABLES viii

LISTOF FIGURES..x

CHAPTER

1. INTRODUCTION……………………………………………………………………1

Overview……………………………………………………………….1

Illustrative scenario ……………………………………………………3

Thesis layout………………………………………………………...4

2. CURRENT RESEARCH TRENDS…………………………………………….5

Smart Environments…………………………………………………...5

Types of Sensors used…………………………………………………7

Challenges……………………………………………………………..8

Examples with Physical Test beds……………………………………10

MavHome Project……………………………………………………10

MIT Media Lab and House_n Project……………………………..11

Intel Research Lab……………………………………………………12

Medical Automation Research Center (MARC) ………………...13

Gator Tech Smart Home Project……………………………………14

Other Projects…………………………………………………………14

Temporal Reasoning and Mining………………………………….....16

Summary …………………………………………………………...18

3. "TEMPAL": AN INTRODUCTION 19

Temporal Relations……………………………………………………19

TempAl Definition……………………………………………………21

The Role of TempAl in the MavHome Smart Home Project………22

MavHome Data Collection……………………………………………26

Synthetic Data Collection…………………………………………..29

Summary…………………………………………………………………30

4. "TEMPAL": DETAILED METHODOLOGY……………………………...31

Representing Temporal Relations…………………………………..31

Benefits of Temporal Relations……………………………………….39

Anomaly Detection Using Temporal Relations……………………….41

Enhancing Prediction Using Temporal Relations…………………..47

Association Rule Mining Approach for Enhancing Prediction ……50

Enhancing Prediction by Adding Temporal Relations-based Probabilities to ALZ ……………………………………………………………..54

Data Visualization……………………………………………………….59

Summary…………………………………………………………………61

5. EXPERIMENTAL FINDING AND DISCUSSIONS………………… ……...62

Temporal Relation Formation…………………………………………62

Anomaly detection……………………………………………………..64

Discussion on Anomaly Detection……………………………………68

Enhancing Prediction…………………………………………………..69

Association Rule Mining Approach for Enhancing Prediction …….70

Discussion on Association Rule Mining Approach for Enhancing Prediction ………………………………………………………………...76

Enhancing Prediction by Adding Temporal Relations based Probability to ALZ ……………………………………………………………77

Test Case …………………………………………………………80

Discussion on Enhancing Prediction by Adding Temporal Relations based Probability to ALZ ……………………………………………………...83

Activity visualization using Time Intervals…………………………84

Summary…………………………………………………………..87

6. CONCLUSION AND FUTURE WORK ……………………………………..88

Conclusion…………………………………………………………88

Future work………………………………………………………..91

7. BIBLIOGRAPHY……………………………………………………………...94

APPENDIX

A. MAVLAB…………………………………………………………………105

B. INPUTS, OUTPUTS AND SCREENSHOTS………………………………109

C. ADDITIONAL TESTCASES…………………………………………………...124

LIST OF TABLES

1. Temporal relations representation which includes all thirteen Allen's relations……...20

2. Nine Temporal relations representation which aid anomaly detection ……………….42

3. Prediction aiding Temporal relations …………………………………………………48

4. Parameter settings for experimentation………………………………………………63

5. Sample display of sensor data across various stages of temporal relation formation…63

6. Characteristics of the synthetic and real training datasets for anomaly detection……65

7. Characteristics of the synthetic and real test datasets for anomaly detection…………65

8: Anomaly detection in the test set for the real dataset…………………………………66

9. Anomaly detection in the test set for the synthetic dataset …………………………...66

10. Parameters setting for training set for prediction experiment………………………..69

11. Parameters setting for test set for prediction experiment……………………………70

12. Parameter settings and rules generated using Apriori-type algorithm in Weka for real dataset……………………………………………………………………………………70

13. Parameter settings and rules generated using Apriori-type algorithm in Weka for synthetic dataset…………………………………………………………………………71

14. Display of a sample of best rules generated…………………………………………72

15. Comparing ALZ based prediction with and without temporal rules in real datasets...74

16. Comparing ALZ based prediction with and without temporal rules in synthetic datasets……………………………………………………………………………………...74

17. Dataset descriptions of training and test set used for experimentation……………...77

18. Comparing accuracy of prediction techniques using TempAl on real datasets……..78

19. Comparing accuracy of prediction techniques using TempAl on Synthetic datasets...78

10. Training, Test Set, Temporal relations for test case scenario…...…………………...81

21. Interesting patterns found using the pattern recognition tool………………………..86

22. Number of patterns found in real experiment dataset………………………………..86

LIST OF FIGURES

1. Temporal intervals are labeled as A “before” B “before” C or A “before” B “finishes-by” C………………………………………………………………………………4

2. Common goals of the smart environment……………………………………………6

3. X-10 Lamp module ……………………………………………………………………7

4. Overview of Argus master slave system ………………………………………………8

5. Current challenges in smart environments……………………………………………...9

6. Current trends in smart environment and intelligent systems research at MIT House_n project, MavHome project and Intel Research …………………………………………10

7. MavHome software architecture ……………………………………………………...23

8. Architecture Overview of TempAl……………………………………………………26

9. MavLab Argus Sensor Network ……………………………………………………...28

10. MavLab Argus and X-10 sensor network capturing the data ………………………29

11. Illustration of time Intervals between three devices television, fan, and lamp………32

12. X before Y………………………………………………………………………………33

13. X after Y……………………………………………………………………………….33

14. X during Y……………………………………………………………………………...34

15. X contains Y…………………………………………………………………………34

16. X overlaps Y……………………………………………………………………………35

17. X overlapped-by Y……………………………………………………………………..35

18. X meets Y………………………………………………………………………………36

19. X met-by Y……………………………………………………………………………..36

20. X met-by Y……………………………………………………………………………..36

21. X started-by Y…………………………………………………………………………..37

22. X finishes Y……………………………………………………………………….37

23. X finished-by Y……………………………………………………………………37

24. X equals Y…………………………………………………………………………38

25. The benefits of temporal relations with examples…………………………………..40

26. Trie formed by the ALZ parsing of the sequence aaababbbbbaabccddcbaaaa………56

27. The steps involved in the processing of temporal relations formulations in datasets..64

28. Anomaly detection on test sets of real and synthetic data in 3-D. ………………….68

29. Association rule mining in real datasets……………………………………………..71

30. Association rule mining in synthetic datasets………………………………………72

31. Percentage accuracy vs. percentage error in real datasets in prediction using association rule mining……………………………………………………………………..75

32. Percentage accuracy vs. percentage error in synthetic datasets in prediction experiment using association rule mining……………………………………………….75

33. Percentage accuracy in real datasets in prediction experiment using Alz and TempAl……………………………………………………………………………………..79

34. Percentage accuracy in synthetic datasets in prediction experiment using Alz and TempAl ……………………………………………………………………………………79

35. Raw output on the test case dataset using Alz…………………………………………………………………………………82

36. Raw output on the test case dataset using Alz and TempAl……………………………………………………………………………………83

37. Interval Visualization Tool……………………………………………………….85

This thesis is dedicated to my Mother and Father.

CHAPTER ONE

INTRODUCTION

Overview

The problems of representing, discovering, and using temporal knowledge arise in a wide range of disciplines, including computer science, philosophy, psychology, and linguistics [1]. Temporal rule mining and pattern discovery applied to time series data has attracted considerable interest over the last few years [2]. We consider the problem of learning temporal relations between event time intervals in smart home data, which includes physical activities (such as taking pills while at home) and instrumental activities (such as turning on lamps and electronic devices) and their results can be used to enhance prediction and to detect anomalies. The purpose of this work is to identify interesting temporal patterns in order to improve prediction of events based on observed temporal relations in a smart home environment and 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.

By 2040, a projected 26% of the US population will be 60+ and at least 45% of the populations of Japan, Spain and Italy will be 60 or older by then. Approximately 13% of these older adults suffer from dementia and related disabilities [3]. Given the costs of nursing home care and the importance residents place on remaining in their current residence as long as possible, use of technology to enable residents with cognitive or physical limitations to remain in their homes longer should be more cost effective and promote a better quality of life. Thus we see a strong need for smart homes in the near future. As a long-term outcome of this investigation we expect to develop and to offer the community smart environment technologies with data mining and machine learning algorithms that can effectively perform a variety of health monitoring and intervention strategies.

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 event prediction and 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. Particularly people with disabilities, elder adults and chronically ill residents 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.

Researchers have different views on how to structure temporal data and how to go about mining temporal information. Each strategy can reflect different a perspective of the problem [1]. We propose one such framework to derive temporal rules from a time series representation of observed resident activities in a smart home, and validate the algorithm using both synthetic datasets and real data collected from the MavHome smart environment. This framework is based on Allen’s temporal logic [1]. Allen suggested that it was more common to describe scenarios by time intervals rather than by time points, and listed thirteen relations formulating a temporal logic (before, after, meets, meet-by, overlaps, overlapped-by, starts, started-by, finishes, finished-by, during, contains, equals) [1]. These temporal relations play a major role in identifying temporal activities which occur in a smart home. 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 enhance prediction and 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 and enhance sequential prediction 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 synthetic data.

The outcome of this research is a new algorithm for anomaly detection and event prediction, called TempAl. TempAl differs from earlier work by incorporating temporal relation representation and discovery into the algorithm. The resulting algorithm represents a contribution which is integrated into the MavHome smart home system and can also be used as a standalone method for temporally-enhanced data analysis.

Illustrative Scenario

In this section we illustrate instances of temporal relations [4], then illustrate as how they can be used for anomaly detection and prediction. Consider a scenario containing three events: A (turn on range top), B (turn on oven), and C (turn on toaster). Figure 1 represents the relationship among the three events A, B, and C. Note that Event A occurs before Event B and Event B occurs before Event C. We can see that A "before" B "before" C is a possible relationship label. However, an alternative representation consistent with the events is A "before" B; B "finishes-by" C. The second interpretation, or relationship, actually changes our perspective of the scenario. In this case when we use the relation B "before" C we know that the event B just occurs before C. In contrast, when we interpret the relationship as B finished-by C, an anomaly can be flagged in cases where B and C do not finish at the same time. Thus the relation of B "finished-by" C is a better fit for the relationship illustrated in Figure 1 between events A, B, and C. Here we see an illustration of the temporal relation.

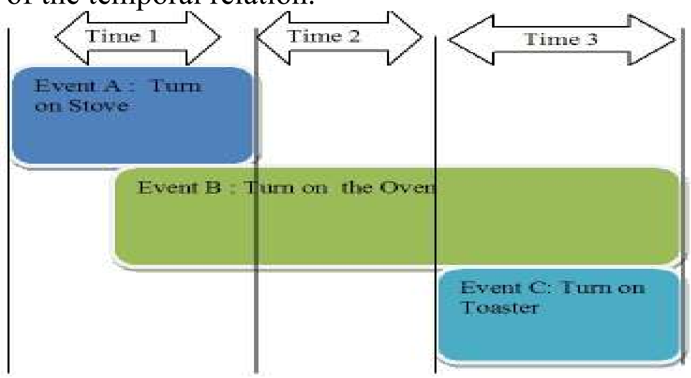


Figure 1. Temporal intervals are labeled as A "before" B "before" C or A "before" B "finishes-by" C.

Consider a simple scenario where these temporal relations play a vital role in anomaly detection. Consider a case where an elderly person takes pills after eating food. We notice that these two activities, taking pills and eating, share the temporal relation "after" between them. When this relationship is violated, the relationship is updated to "meets" and an anomaly in activity is noted. Similarly, temporal relations can enhance the smart home's ability to predict events. In our scenario, the algorithms can predict the taking of pills after consuming food.

Thesis Layout

In this thesis we start by introducing the problem and approach in chapter one and current research trends in chapter two. In chapters three and four, we discuss our methodology. In chapter five we the results of our experimental validation and in the final chapter we conclude with observations and talk about possible future work.

CHAPTER TWO

CURRENT RESEARCH TRENDS

The research described in this thesis contributes toward the emerging domains of smart environments or smart homes. In this chapter we summarize recent advances in smart environment research and current trends in temporal relations-based data mining and knowledge discovery.

Smart Environments

Mark Weiser gave his view of ubiquitous computing as the following:

> "A physical world that is richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives, and connected through a continuous network"
>
> \- *Mark Weiser* [5]

We define a smart environment as *a small world where all kinds of smart devices are continuously working to make residents' lives more comfortable.* Smart environments aim to satisfy the experience of residents in every environment, by replacing the hazardous work, physical labor, and repetitive tasks with automated agents [6] and also ensure security, comfort and health & well-being of the resident. The general features which are incorporated into most smart environments include home automation such as remote control of devices, inter-device communication, information acquisition using sensors, enhanced services using intelligent devices, and task automations using prediction techniques and data mining algorithms [6]. Smart environment research efforts are by nature multi-disciplinary projects which make use of advances in wireless communication, databases, algorithm design, speech recognition, image processing, computer networks, mobile computing, ubiquitous computing, tele-health, operating systems, assistive technologies, adaptive controls, sensor designs, software engineering, middleware architectures, parallel processing, pervasive computing, and ambient intelligence [5].

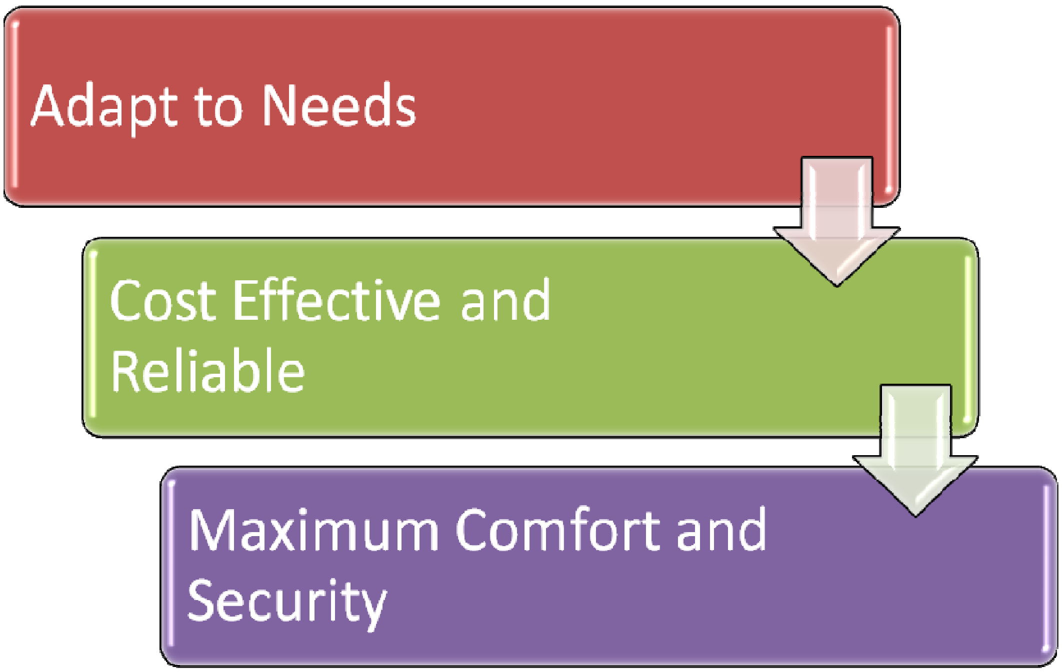


Figure 2. Common goals of the smart environment.

Common goals of smart environments include adapting to the needs of residents, providing services which are cost effective and reliable, and providing maximum comfort and security to the resident. The contributions that have been offered by smart environment research projects are the design and implementation of interfaces, applications, and systems ranging from motion detection sensors to device automation in homes, which can be used by residents, anytime [7].

Types of Sensors used

The sensors used for our data collection mainly consist of an X-10 sensor network and an Argus sensor network. We have many X-10 sensor systems available in stores today. In our environment, we have specifically used RF transceivers, computer interface modules, light modules, appliance modules, motion detectors, and an HVAC thermostat. Figure 3 illustrates a simple x-10 lamp module [7] [8].

Figure 3. X-10 Lamp module [8].

Environment events are noted by the X-10 sensors, and are sent through the power line to an awaiting receiver. We note that the other part of the data collection sensor consist of the Argus sensor network which are devices that operate off of the software stored on chip. This Argus sensor system consists of slaves and dongles which form the Master-Slave network for sensory reception. For more information on the smart home layout and sensor deployment, see Appendix A. The Figure 4 illustrates the Argus sensor networks master slave system.

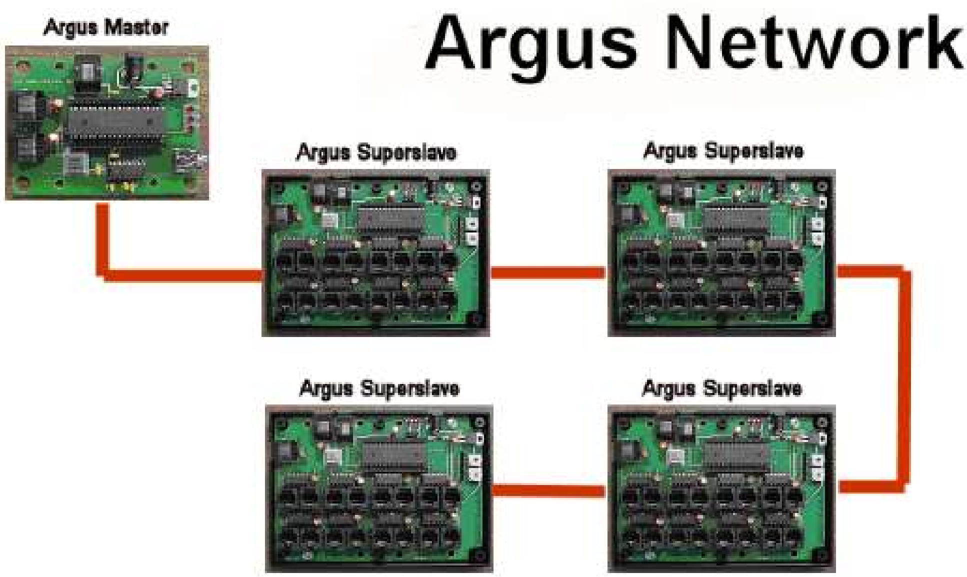


Figure 4. Overview of Argus master slave system [8].

Challenges

Current challenges in smart environments today include not only the need for innovative, user-friendly applications and techniques but also large amounts of interventions to setup, maintain and upgrade the environment, with new sensors, technologies and applications which suits out needs. We desire technologies which become a part of our everyday life and dissolve into our life to the point where they become unnoticeable but significantly improve our life and the way we lead it. Researchers are investigating the intelligent environment frameworks that could recognize natural human behaviors, interpret and react to these behaviors, and adapt to residents in a non-intrusive manner. These features of an intelligent environment present difficult challenges to solve. Another challenge is to seamlessly integrate different fields of study and research such as computer science, digital devices, and wireless and sensor networking to create an intelligent environment. Some current challenges which are being explored are illustrated in Figure 5. These challenges belong to the domains of smart devices (Intelligent devices), virtual pets, human-computer interaction, healthcare, sensors networks, learning and adaptation to users and their lifestyles.

Figure 5. Current challenges in smart environments (smart devices, robotics, HCI, healthcare, sensors & communication, and learning & adaptation).

Examples with Physical Test beds

With the convergence of supporting technologies in artificial intelligence and pervasive computing, smart environment research is quickly maturing. The goals of intelligent systems are to reason, predict, and make decisions that will automate a person's physical environment (e.g., home, workplace, and so forth) in a way that adapts to the resident's life style and makes the environment more supportive. Figure 6, illustrates some significant current projects being pursued in the research world.

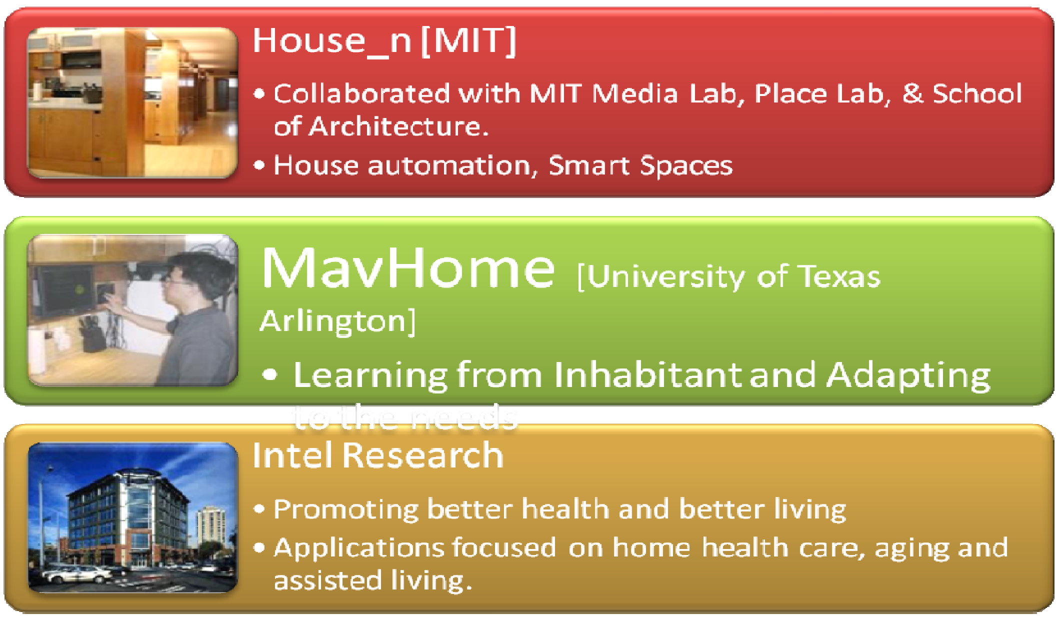


Figure 6. Current trends in smart environment and intelligent systems research at MIT House_n project, MavHome project and Intel Research [9] [10] [11].

MavHome Project

The MavHome project treats an environment as an intelligent agent, which perceives the environment using sensors and acts on the environment using powerline controllers [11]. At the core of its approach, MavHome observes resident activities as noted by the sensors. These activities are mined to identify patterns and compression-based predictors are employed to identify likely future activities [12]. Some current challenges in this project are better human-computer interactive applications, healthcare focus, advanced sensor systems and new algorithms for learning and adapting to residents of a smart environment including new parameters such as space and time.

Application of MavHome algorithms to healthcare includes anomaly detection, on health datasets to check for outliers and drifts in smart homes [14]. This approach is based on regression and correlation on numerical-based health datasets and would not apply to activities which consist of devices or actions, for instance, turning on and off of devices in smart home. Furthermore, this approach considers each event is occurring in a single instant, and therefore overlooks the time interval encompassed by an event. As a result, there is a need to design a more effective and more general anomaly detection model. Prediction and decision making has experienced significant success and could automate a resident's activities, but this can be improved using time as a component. Currently this project is looking towards new sensor systems and trying to address the problem of multiple residents [13] [15] [16] [17] [18].

MIT Media Lab and House_n Project

The MIT Media Lab is focused on gadget creation and specific implements of the future [20]. Many of these projects could be incorporated into an intelligent environment to enhance the resident's experience, but they probably will not be commercially available for another decade. The work in this thesis does not incorporate any MIT Media Lab technology primarily due to their availability and the significant amount of engineering effort that would be required to duplicate and integrate their work; however, specific ideas such as those in the augmented reality kitchen, localized context awareness, and the interactive nature of many of their projects could be incorporated into our environments.

The Place Lab developed by the MIT House_n Consortium and TIAX, LLC currently is researching methods to validate performance of the activities of daily life and biometric monitoring. The rich sensing infrastructure of the Place Lab is being used to develop techniques to recognize patterns of sleep, eating, socializing, recreation, etc. Particularly for the elderly, changes in baseline activities of daily living are believed to be important early indicators of emerging health problems – often preceding indications from biometric monitoring [21]. There work on recognition of Activities of Daily Living in the Home Setting using Ubiquitous, Sensors when applied with pattern classification and context-based AI algorithms which involve time series based models can be considered [22].

Another group at MIT, called the Agent-based Intelligent Reactive Environments group (AIRE) [23] conducted research on pervasive computing and people centric applications to construct intelligent spaces or zones. Their work included an intelligent conference room, intelligent workspaces, kiosks, and oxygenated offices.

Intel Research Lab

Intel Corporation’s Proactive Health Lab is exploring technologies to help seniors “age in place” in order to help the increasing health care burden of the rapidly aging population of the United States by anticipating resident needs through observation with wireless sensors and taking action to meet those needs through available control and interactive systems.

The goal of the Computer-Supported Coordinated Care (CSCC) project [24] at Intel Research is to identify the characteristics and needs of the care networks for elders who wish to remain at home ("age in place"). Ultimately, their goal is to develop technology to help this population. In a three-phase study towards this end, they developed an empirical approach focused on the wide range of people involved with home elder care [25] [26]. Response time and more generally using time as a parameter is an important factor for most healthcare system, though there current work involves empirical approaches; data mining models should also be investigated.

Medical Automation Research Center (MARC) Smart House Project

The Medical Automation Research Center (MARC) smart house project [27] at the University of Virginia is focused on the issue of in-home monitoring for the elderly in order to promote the concept of aging in place. Their in-home monitoring system is made up of low-cost, non-invasive sensors (without cameras or microphones) and communications to establish Telematics to authorized residents (for example, family, personal physician).

MARC is designed to perform health status monitoring by analyzing behavioral patterns of its residents using collected metrics (Barger et al. 2003). The data logged is used to observe general health and activity levels and using data mining techniques such as analysis of mixture models to monitor what is called the Activities of Daily Live (ADL) [28]. ADL also includes the measure of the index of well-being and a measure of the decline in ability over time. The data analysis component uses Estimation Maximization (EM) algorithms and Mixture Models (MM) to yield unique health status reports that can be made available to the residents, their medical advisors and family members. Monitoring ADLs can also be beneficial an as early indicators for an onset of a disease. Moreover, their system provides identified activity levels, which could lead to reality-based decision making. Such a system would be beneficial if it were used to evaluate the quality of the day that a person could have, based on the previous observed activity levels, and suggest required changes and modifications in the daily activities patterns which would lead the resident to experience a better quality of life (for example, the home perceives that the resident has irregular sleeping patterns and this observations can be used to make corrections and suggestions, which could improve the resident lifestyle and health) [29].

Gator Tech Smart Home Project

The Gator Tech Smart home is built from the ground up as an assistive environment to support independent living for older people and residents with disabilities [30]. Currently, the project uses a self-sensing service to enable remote monitoring and intervention for caregivers of elderly persons living in the house. Their current key contribution is the development of a middleware architecture which includes a physical layer of devices, a sensor platform layer to convert readings into service information, a service layer to provide features and operators to components, a knowledge layer that offers ontology and semantics, a context management layer to provide context information, and an application layer to support a rich set of features for resident living. The state of the project is still focused on integration and the middleware development, but they are beginning to focus on issues with eldercare and the aging in place initiatives [31].

Other Projects

There are also a number of systems which have been developed to help people compensate for physical and sensory needs. We see that most of them rely on computer based technologies incorporating artificial intelligence techniques (for example, schedule management using the Autominder system) [32].

A schedule management system for the elderly helps people who suffer from memory decline—an impediment that makes them forget their daily routine activities such as taking medicine, eating meals, or personal hygiene. Autominder [32], an intelligent cognitive orthotic system for people with memory impairment, employs techniques such as dynamic programming and Bayesian learning, a web-based interface for plan initialization and update to construct rich models of a resident's activities—including constraints on the times and ways in which activities should be performed to monitor the execution of those activities, detect discrepancies between what a person is expected to do and what he or she actually is doing, and to reason about whether to issue reminders [33]. Assistive technologies, when combined with the monitored information on daily activities of the resident, can be used to measure the quality of a person's performance of their daily routine activities. A schedule management system such as this could generate an improved resident lifestyle based on behavioral patterns designed to improve their daily performance [34].

An extended application of anomaly detection is its use for reminder assistance. Autominder, an intelligent cognitive orthotic system for people with memory impairment, employs techniques such as dynamic programming and Bayesian learning to remind residents about their planned Activities for Daily Living. Autominder includes a web-based interface for plan initialization and constructs rich models of a resident's activities—including constraints on the times and ways in which activities should be performed—to monitor the execution of those activities. Autominder looks for differences between expected and observed activities, and reasons about whether to issue reminders.

The University of Essex's intelligent dormitory (iDorm) is a real ambient intelligent test-bed comprised of a large number of embedded sensors, actuators, processors and networks in the form of a two bed roomed apartment. Fuzzy rules are learned from the observed resident activities [35] and are used to control select devices in the dorm room.

The goal of the Point-of care Lab at the Oregon Health & Science University (OHSU) is to develop approaches and technologies that allow early detection and remediation of physical and cognitive decline [36]. Scientists there are creating unique artificial intelligence algorithms that combine information from a variety of sensors and tracking devices placed throughout the homes of seniors, to assess situations in which mobility or cognition problems may be occurring, and to provide intervention and health coaching to seniors to assure their health care needs are being met. Such systems can enhance their performance and improve accuracy, if time models designed with temporal data mining techniques are considered.

Temporal Reasoning and Mining

Temporal mining is a reasonably 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. Some work on temporal data reasoning and mining has been done in the context of classical and temporal logics and have been applied to real-time systems to artificial intelligence projects. In this section, we give a general overview of some current research trends in temporal reasoning and mining.

Morchen argued that Allen's temporal patterns are not robust and small differences in boundaries lead to different patterns for similar situations [37]. Morchen presents a Time Series Knowledge Representation (TSKR), which expresses the temporal concepts of coincidence and partial order. He mentions that Allen's temporal relations are ambiguous in nature, making them not scalable and not robust. Morchen handles the problem of using ambiguous nature of Allen's relations by applying constraints to define the temporal relations. Although this method appears feasible and computationally sound, it does not suit our smart home application due to the granularity of the time intervals in smart homes datasets. We need to note that the time granularities here indicate the events in smart homes are instantaneous and some of them just occur for long periods and some just occur for a split second short, where as Morchen applies TSKR to muscle reflection motion and other such areas where time intervals are consistently similar in length. His approach does not involve ways to eliminate noise and the smart home datasets are so huge that computational efficiency would not be the only factor to be considered. Morchen also describes the temporal constraints using their description language. Overall they proposed a logic-based approach to describe temporal constraints with multiple time granularities related to events occurring in smart homes. Morchen identified time and sensor granularities as sequences of time points properly labeled with propositional symbols marking the starting and ending points in each granule. Temporal constraints that are modeled describe temporal relationships related to sensors providing the right control of the environment of smart home.

In artificial intelligence, the event calculus is a frequently-used approach for representing and reasoning about events and their effects. Björn, et al. [38] 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 in their work on probabilistic reasoning about uncertain relations between temporal points [39] represent the uncertain relation between two points by an uncertainty vector with three probabilities of basic relations (“<”, “+”, “>”). They also incorporate inversion, composition, addition, and negation operations into their reasoning mechanism. This model would not be suitable for a smart home scenario as it would not go into final granularities to analyze instantaneous events. The work of Worboys, et.al. [40] involves spatio-temporal-based probability models, the handling of which is currently identified as future work. Dekhtyar, et. al.'s work on probabilistic temporal databases [41] provides a framework which is an extension of the relational algebra that integrates both probabilities and time. This work includes some description of Allen's temporal relations and some of these are incorporated already in this current work.

Summary

In this chapter we discussed the current trends in smart environment research and the current trends in temporal reasoning and mining research. Smart homes are intelligent environments which are designed used to make the life of residents easier and aid them in everyday activities. In the next chapter we will present related work on temporal relations and give an introduction to our work.

CHAPTER THREE

"TempAl" (TEMPORAL ANALYZER): AN INTRODUCTION

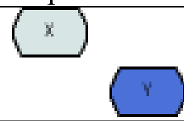
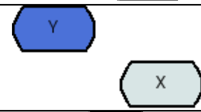
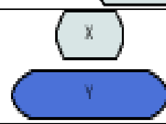
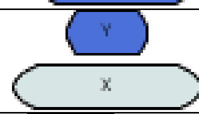
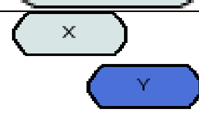
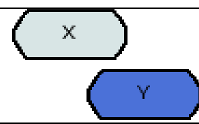
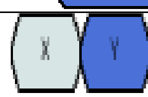
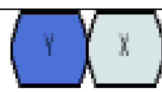
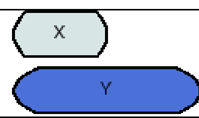
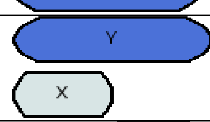
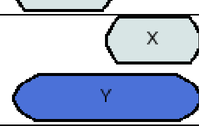
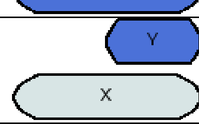
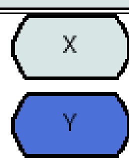
In this chapter we introduce the concept of temporal relations and introduce principles for representing and reasoning about temporal relations. In addition, we give an architectural overview of our analyzer and provide an overview of our data collection environment.

Temporal Relations

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 coffeemaker, and so forth. Instrumental activities are those which have some interaction with an instrument which is present and used in a home. 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 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 can act as reminder assistants and help detect anomalies and aid us in taking preventive measures.

Allen listed thirteen relations (visualized in Table 1) comprising a temporal logic: before, after, meets, meet-by, overlaps, overlapped-by, starts, started-by, finishes, finished-by, during, contains, and equals [42]. These temporal relations play a major role in identifying time-sensitive activities which occur in a smart home. Consider, for instance, 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

Table 1: Temporal relations representation which includes all thirteen Allen's relations.

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)

Modeling temporal events in smart homes is an important problem and offers advantages to residents of smart homes. We see that the temporal constraints can model causal activities; if a temporal constraint is not satisfied then a potential "abnormal" or "critical" situation may have occurred. Similarly, they can be used to form rules which can be used for prediction. For example, if there is a rule which states that there is a large propability of turning on the television after having dinner we can use it to predict turning on the television to be the next event and use this prediction to automate the turning on of the television after dinner.

TempAl Definition

"TempAl" (Pronounced as "temple") is a suite of software tools which enrich smart environment applications by incorporating temporal relationship information for various applications including prediction and anomaly detection. In smart homes, the time when an event takes place is known and is recorded. The previous model in our smart home did not incorporate time for analysis purposes. We felt that including this information would improve the strength of the smart home algorithms, which motivated our contributions of storing, representing, and analyzing timing information. The temporal nature of the data provides us with a better understanding of the nature of the data. We see that using a time series model is a common approach to reasoning about residents time-based events. However, we consider events and activities using time intervals rather than time points, which is appropriate for home scenarios [43]. Thus we have designed a novel approach to solve the problem of incorporating time for various smart home applications. We introduce the notion of temporal representation which is capable of expressing the relationship between interval-based events. We develop methods for finding interesting temporal patterns as well as for performing anomaly detection and prediction based on these patterns.

The contribution of this work is a new means of temporal representation for smart home activities and events which help with reasoning-related tasks, including planning, explanations and predictions. Our focus for this thesis is on anomaly detection and prediction. Figure 7 provides an architectural overview of this tool kit. In this Figure, we see the various components which comprise to form TempAl. We also propose a model to enhance prediction and a simple evidence-based anomaly detection model for computing whether the current event is anomalous or not in the chapter 4. The results for these experimentations are reported in chapter 5.

The Role of TempAl in the MavHome Smart Home Project

The existing system MavHome architecture contains the software components ProPHeT [44], Ed [45], ALZ [46], and Arbiter [47] [8], as illustrated in Figure 7. Inside this system framework exists the core system architecture for our approach. We present the architecture of "TempAl" which enhances the Active LeZi predictor shown in Figure 7. In this section we outline the components we utilize in our work and place those in a framework. We will now present the specific core architectural elements which we utilize to enhance the current architecture, and it will be analyzed in this chapter. The goals of our system are to learn a model of the inhabitants of the intelligent environment, automate devices to the fullest extent possible using this model in order to maximize the comfort of the inhabitant while maintaining safety and security, and adapt this model over time to maintain these requirements. In order to accomplish these goals, we must first learn a model of inhabitant activities, and then incorporate this into an adaptive system for continued learning and control.

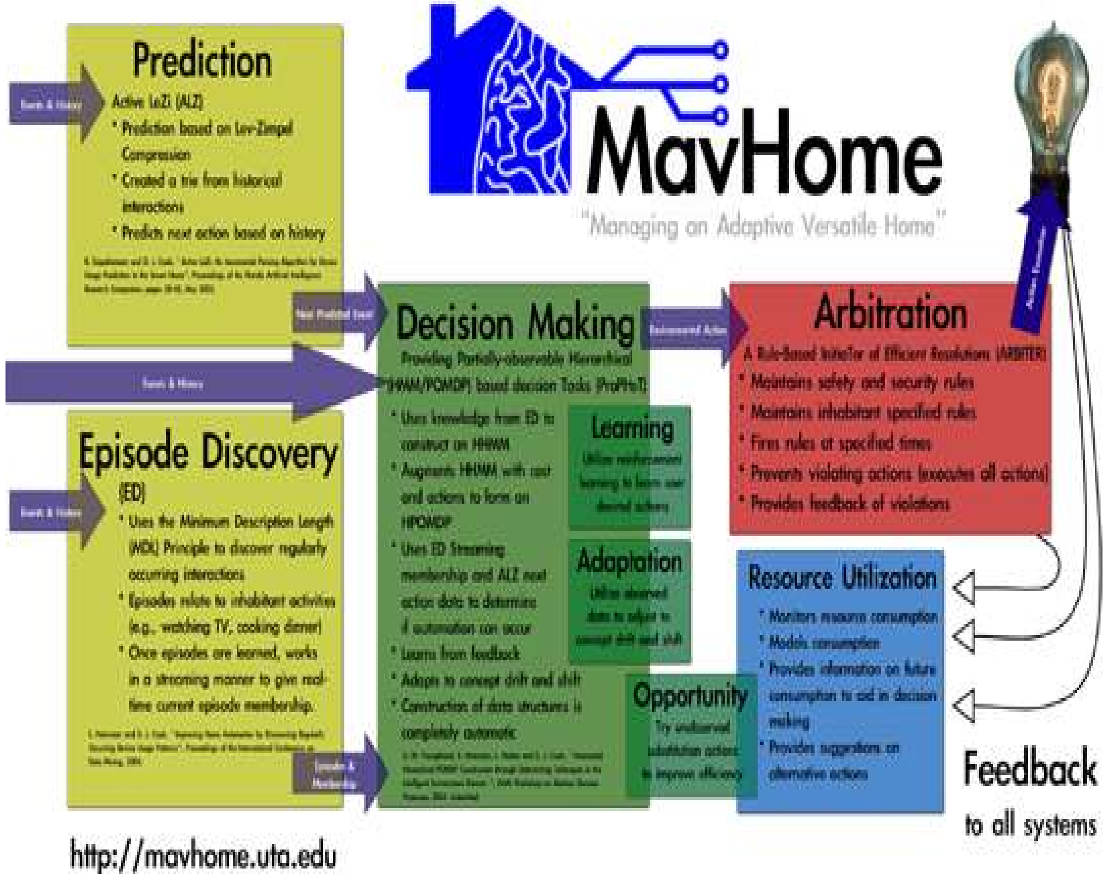


Figure 7. MavHome software architecture [8].

ProPHeT

Decision making is performed in the ProPHeT (Providing Partially-observable Hierarchical (HMM/POMDP) based decision Tasks) component [44]. The world representation at this level is the Hierarchical Hidden Markov Model (HHMM) [8] based upon a hierarchy of episodes of activity mined from stored observations. Episode Discovery (ED) [45] is used to generate low-level episode Markov chains and build the hierarchy of abstract episodes under the direction of ProPHeT. Learning is performed by extending the HHMM to a hierarchical Partially Observable Markov Decision Process (HPOMDP) and applying temporal-difference learning [12].

Episode Discovery (ED)

The Episode Discovery (ED) data-mining algorithm [45] [47] [8] discovers interesting patterns in a time-ordered data stream. ED processes a time-ordered sequence, discovers the interesting episodes that exist within the sequence as an unordered collection, and records the unique occurrences of the discovered patterns.

Active LeZi (ALZ)

An intelligent environment must be able to acquire and apply knowledge about its residents in order to adapt to the residents and meet the goals of comfort and efficiency. These capabilities rely upon effective prediction algorithms. Given a prediction of resident activities, MavHome can decide whether or not to automate the activity or even find a way to improve the activity to meet the system goals. Specifically, the MavHome system needs to predict the inhabitant's next action in order to automate selected repetitive tasks for the inhabitant. The system will need to make this prediction based only on previously-seen inhabitant interaction with various devices. It is essential that the number of prediction errors be kept to a minimum–not only would it be annoying for the inhabitant to reverse system decisions, but prediction errors can lead to excessive resource consumption. Another desirable characteristic of a prediction algorithm is that predictions be delivered in real time without resorting to an offline prediction scheme MavHome uses the *TDAG Active-LeZi* algorithm (ALZ) [48] to meet the prediction requirements.

ARBITER

When issues of safety and security are of the highest importance in a system there is the need for an enforcer of safety and user preference rules before actions are made. This system works by using a knowledge base of rules and evaluating each action event against these rules to determine if the action violates them. Before an action is executed it is checked against the policies in the policy engine, ARBITER (**A** **R**ule-**B**ased **I**nitia**T**or of **E**fficient **R**esolutions). These policies contain designed safety and security knowledge and resident standing rules [8].

TempAl

TempAl, also known as temporal analyzer is a suite of tools which are used for identifying the temporal information in smart home activities, and use this information for the process of anomaly detection in smart home activities and also enhance the prediction of activities. TempAl's prediction component is an extension to the ALZ based predictor. We see that, TempAl uses the raw sensor data and parses it to identify time intervals, using the constraints described in Table 1; TempAl forms the temporal relations which can be saved into a database, but currently is a text file, which later are used by the anomaly detection or the prediction components. This model can also apply to online data or live streaming data, and thus making this applicable to dynamic world. The basic architecture is illustrated in Figure 8. This architecture gives us an overview of the tools which together form TempAl. We see that the raw data is read and parsed by a parser to identify interval data, which is later read by a temporal relations formulation tool, which associates all the time interval data to form temporal relations data. This temporal relations data is later used by the anomaly detection component or the prediction enhancing component, for achieving their goals and their basic functioning.

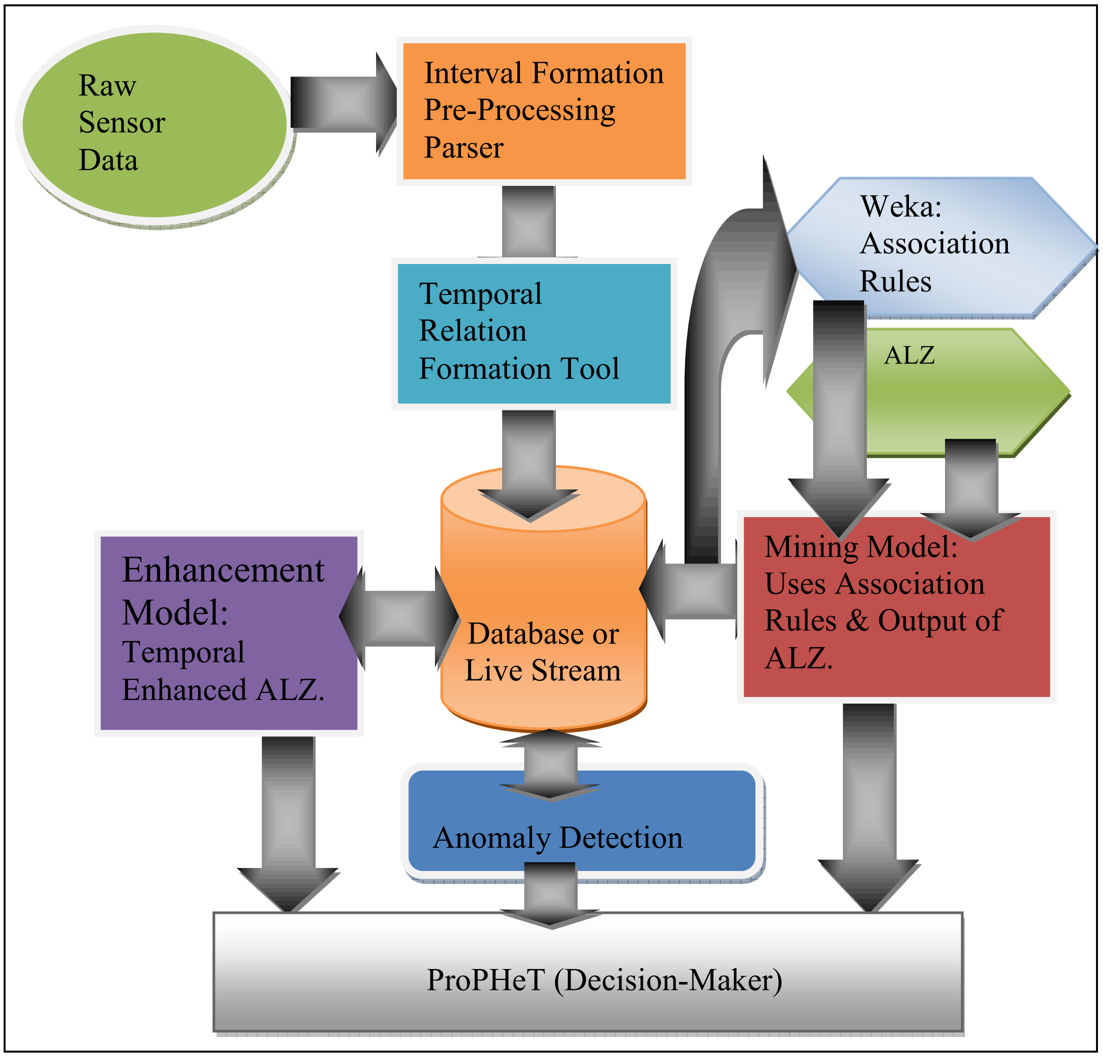


Figure 8. Architecture Overview of TempAl.

MavHome Data Collection

The sensors present in smart environments provide us information about the actions, events and activities happening in the smart space. Every time an event occurs the corresponding sensor or device provides information about the current state it is in and the timestamp as when that information was observed, read or collected.

The algorithms described here are part of the MavHome multi-disciplinary project, which has been engaged in the creation of adaptive and versatile home and workplace environments in the past few years [49]. The goal of the MavHome project is to create a smart home that can act as an intelligent agent. The home perceives the state of the environment and its residents using sensors, reasons about the state and possible actions using machine learning algorithms, and acts on the environment using power line controllers. In order to design a smart environment, we need to design machine learning algorithms that can identify, predict, and reason about resident behaviors. The objective of our initial MavHome study was to determine if our algorithms could learn an automation policy that would reduce the number of manual interactions the resident performed in a smart environment. Our machine learning algorithms did accurately predict resident activities and substantially reduce the average number of daily manual interactions [8] [48].

The MavHome algorithms are tested in two physical environments. One is a smart apartment called the MavPad and another is a smart workplace environment, the MavLab. Our experiments are based on two months of real activity data collected in the MavLab working environment. During that time, a student volunteer performed his normal daily work activities in this environment. All interactions with lights, blinds, fans, and electronic devices were performed using X10 controllers, so that all sensor and interaction events could be captured in a text file. The layout of sensors and controllers in the MavLab is shown in Fig. 9. The data collection system consists of an array of sensors and X10 power line controllers, connected using an in-house sensor network. As shown in Fig. 8, MavLab consists of a presentation area, a kitchen, student desks, a lounge, and a faculty room. There are over 100 sensors deployed in the MavLab that include motion, light, temperature, humidity, and reed switches.

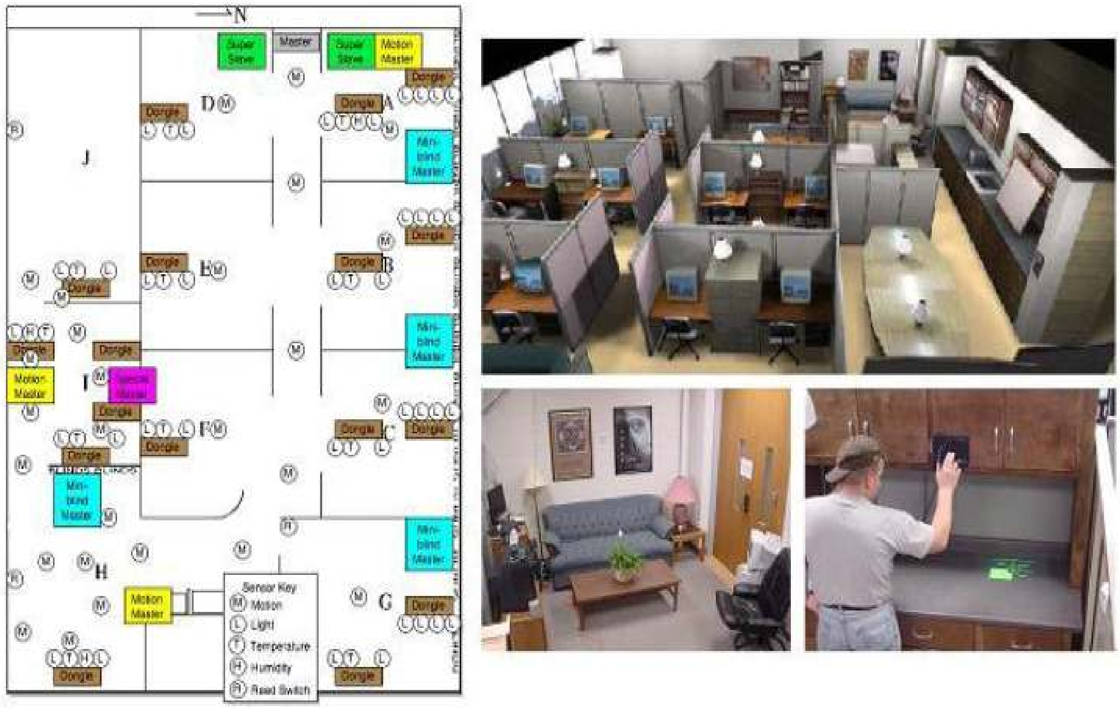


Figure 9. MavLab Argus Sensor Network (M-Motion sensor, L-Light sensor, T-Temperature sensor, H-Humidity sensor, R- Reed switch sensor, S-Smoke sensor, C- Gas sensor) [8]. For more information please visit the Appendix section A (information on sensors and there layouts).

The Argus Sensor Network is the main perception system in MavLab. It is built around a core Argus Master Board. The Master in combination with Super slaves (separate component boards that relay sensor information to a Master board) and Dongles (separate component boards that host up to four sensors and connect to Super slaves) form the Argus Master-Slave (ArgusMS) network. The master is also extended into Argus and Argus Motor (ArgusMS) networks. Argus D allows only pure digital I/O form the network and ArgusMS allows control of stepper motors. An X-10 powerline/based controller is used to monitor electrical outlet usage, light usage and the overhead fan. Refer to Appendix A and B for the layout and other information. A simple illustration of how the data is collected is illustrated in Figure 10.

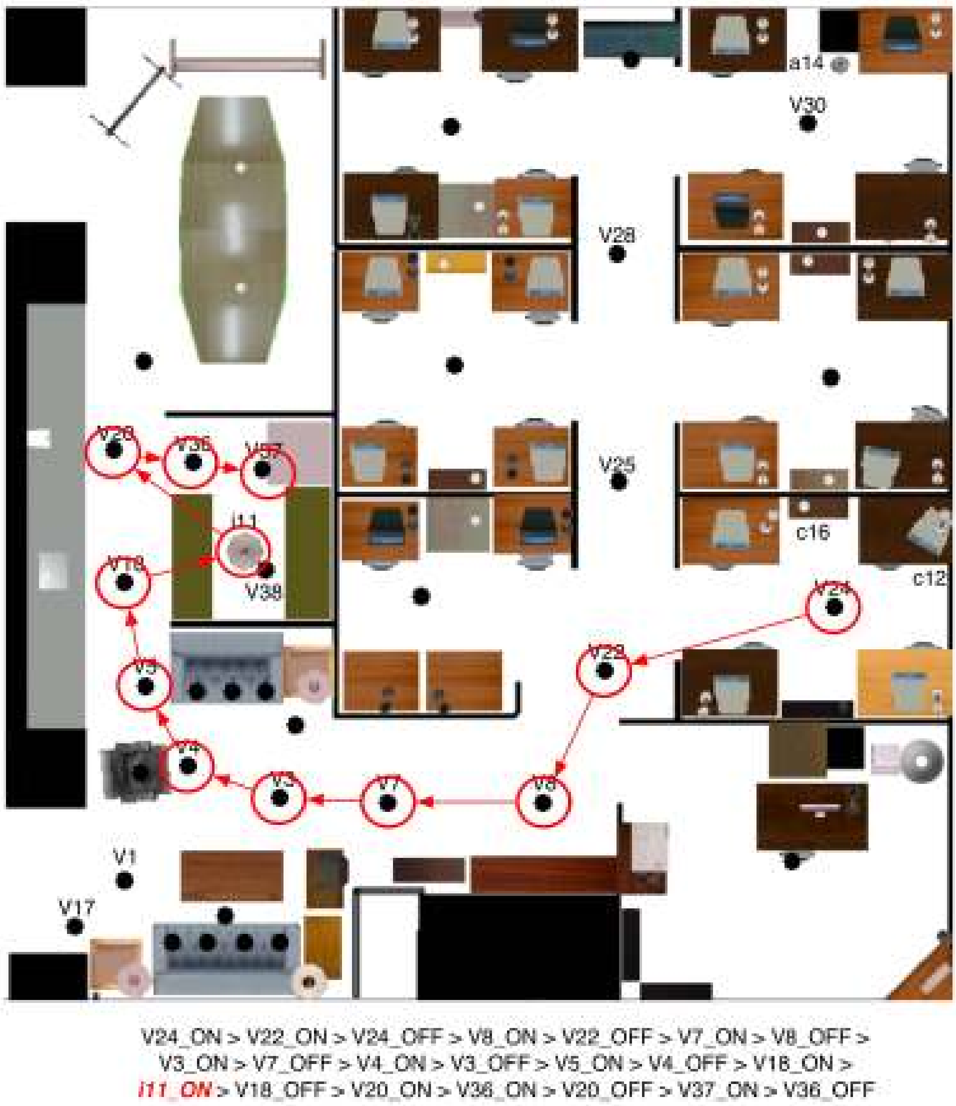


Figure 10. MavLab Argus and X-10 sensor network capture data as the resident takes a break and walks from a desk to the kitchen, opens the refrigerator [8].

Synthetic Data Collection

In addition, we created a synthetic data generator to validate our approach. The data generator allows us to input event sequences corresponding to frequent activities, and specify when the sequences occur. Randomness is incorporated into the time at which the events occur within a sequence using a Gaussian distribution. We developed a model of a user's pattern which consists of a number of different activities involving three rooms in an environment and eight devices. Our synthetic data set contains about 4,000 actions representing two months of activities. The synthetic data consist of 8 devices which can be considered to be spread across a virtual environment of two rooms and simulates these device activities for a sixty day period. For more information on the synthetic data generation outputs, please visit Appendix B.

Summary

In this chapter we introduced temporal relations and gave an overview of the TempAl architecture. We also gave a description of the environment in which we collected data for our algorithm validation. The next chapter will include more details on the steps involved in capturing, reasoning about, and using temporal relationship information in smart environments.

CHAPTER FOUR
"TempAl" (TEMPORAL ANALYZER): DETAILED METHODOLOGY

In this chapter we discuss the methodologies involved in "TempAl" in detail. We also talk about temporal relations and their benefits. Two algorithms which can benefit from temporal analysis, anomaly detection and prediction, are described here. In addition, we present a visualizer which includes a pattern search mechanism to identify interesting temporal patterns spanned across a single-day window size.

Representing Temporal Relations

"It is common to describe scenarios using time intervals rather than time points"

- James F. Allen [43]

The relative positioning of two intervals can be described using Allen's temporal relations. These relations are commonly used for formulating temporal rules involving intervals [50]. The ability of providing and relating temporal information Allen listed thirteen relations 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 temporal activities which occur in a smart home. Consider, for instance, 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. Therefore, when the relationship is violated, an anomaly is noted.

Let us consider a scenario which involves a television, a fan and a lamp being used in a smart home. We see that the resident turns on the television and after some period of time turns on the fan. As time progresses, feeling cold, the resident turns the fan off and the resident continues watching the television. Later on, the television is turned off and the resident turns on the lamp to illuminate the room. We see that this scenario involved three activities each defined by interacts with a single device, namely a television, a fan and a lamp. Now we apply Allen's logic to establish the temporal relations among the activities which occurred. The scenario is illustrated in Figure 11. These activities can be represented as television "contains" fan and "meets" lamp. We can also represent these relationships as television "meets" lamp and fan "before" lamp.

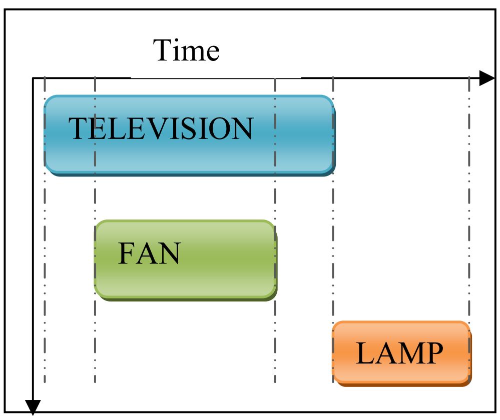


Figure 11. Illustration of time Intervals between three devices television, fan, and lamp.

Modelling temporal events in smart homes is an important problem and offers great advantages to people with disabilities and the elderly. We see that temporal constraints can model normal activities; if a temporal constraint is not satisfied then a potential "abnormal" or "critical" situation may occur. The goal of this experiment is to identify temporal relations in smart home datasets and later use them for health monitoring; specifically for event prediction and anomaly detection

We identify two major problems associated with using Allen's temporal relations. The first problem is the failure of Allen's approach to identify a single most descriptive relation between a pair of events. The second challenge is how to process event relationships in smart home data, which by its nature has a minute time granularity. In our implementation we try to resolve these problems and provide an alternate solution as to how the temporal relations can be identified and analyzed in smart home datasets.

The best way to eliminate ambiguity in identifying the temporal relations is to identify and define the boundary conditions for the thirteen defined intervals before we use it in our algorithm. We illustrate these boundary conditions, using events X and Y as example events. The illustrations are represented by Figures 12 to 24.

X before Y**:** The relation "before" is used between two events X & Y, when the event X occurs before event Y and satisfies the constraint that StartTime(X) is less than StartTime(Y) and EndTime(X) is less than StartTime(Y). This is illustrated in Figure 12. Constraint:-StartTime(X) <StartTime(Y) & EndTime(X) <StartTime(Y).

Figure 12. X before Y.

X after Y: The relation "after" is used between two events X & Y, when the event X occurs after the event Y and satisfies constraint that Start Time(X) is greater than StartTime(Y) and EndTime(Y) is less than StartTime(X). This is illustrated in Figure 13. Constraint: - StartTime(X)>StartTime(Y); EndTime(Y) < StartTime(X).

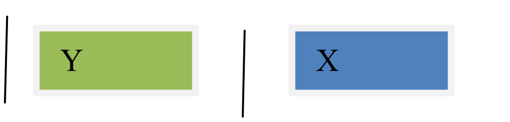


Figure 13: X after Y.

X during Y: The relation "during" is used between two events X & Y, when the event X occurs during the event Y and satisfies the constraint that StartTime(X) is greater than StartTime(Y) and EndTime(X) is less than EndTime(Y). This is illustrated in Figure 14.

Constraint: - StartTime(X)>StartTime(Y); EndTime(X) < EndTime(Y)

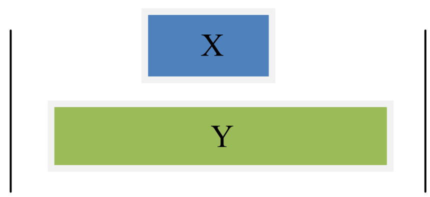


Figure 14: X during Y.

X contains Y: The relation "contains" is used between two events X & Y, when the event X occurs containing the event Y and satisfies the constraint that StartTime(X) is less than StartTime(Y) and EndTime(X) is greater than EndTime(Y). This is illustrated in Figure 15.

Constraint: - StartTime(X) < StartTime(Y); EndTime(X) > EndTime(Y)

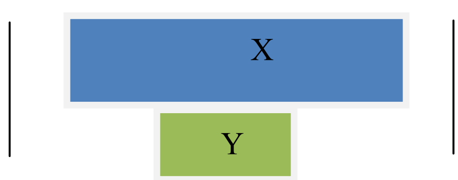


Figure 15: X contains Y.

X overlaps Y: The relation "overlaps" is used between two events X & Y, when the event X occurs overlapping the event Y and satisfies the constraint that StartTime(X) is less than StartTime (Y), StartTime(Y) is less than EndTime(X) and EndTime(X) is less than EndTime(Y). This is illustrated in Figure 16.

Constraint: - StartTime(X) <StartTime(Y); StartTime(Y) <EndTime(X); EndTime(X) < EndTime(Y).

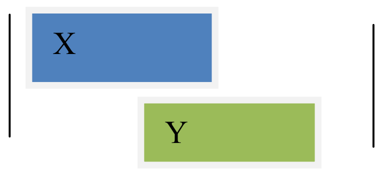


Figure 16: X overlaps Y.

<u>X overlapped-by Y:</u> The relation "overlapped-by" is used between two events X & Y, when the event X occurs overlapped-by the event Y and satisfies the constraint that Start Time(Y) is less than StartTime(X), StartTime(X) is less than EndTime(Y) and the EndTime(Y) is less than EndTime(X). This is illustrated in Figure 17.

Constraint: - StartTime(Y) <StartTime(X); StartTime(X) < EndTime(Y); EndTime(Y) < EndTime(X).

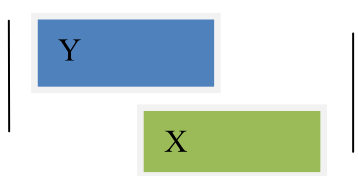


Figure 17: X overlapped-by Y.

<u>X meets Y:</u> The relation "meets" is used between two events X & Y, when the event X occurs meeting the event Y and satisfies the constraint that EndTime(X) is equal to Start Time(Y). This is illustrated in Figure 18.

Constraints: - StartTime(Y) = EndTime(X)

Figure 18: X meets Y.

<u>X met-by Y:</u> The relation “met-by” is used between two events X & Y, when the event X is met-by the event Y and satisfies the constraint that StartTime(X) is equal to EndTime(Y). This is illustrated in Figure 19.

Constraint: - StartTime(X) = EndTime(Y)

Figure 19: X met-by Y.

<u>X starts Y:</u> The relation “starts” is used between two events X & Y, when the event X starts the event Y and satisfies the constraint that StartTime(X) is equal to StartTime(Y) and EndTime(X) is not equal to EndTime(Y). This is illustrated in Figure 20.

Constraint: - StartTime(X) = StartTime(Y); EndTime(X) ≠ EndTime(Y)

Figure 20: X met-by Y.

<u>X started-by Y:</u> The relation “started-by” is used between two events X & Y, when the event X is started-by the event Y and satisfies the constraint that StartTime(Y) is equal to StartTime(X) and EndTime(X) is not equal to EndTime(X). This is illustrated in Figure 21.

Constraint: - StartTime(Y) = StartTime(X); EndTime(X) ≠ EndTime(Y);

Figure 21: X started-by Y.

<u>X finishes Y:</u> The relation "finishes" is used between two events X & Y, when the event X finishes by the event Y and satisfies the constraint that StartTime(X) is not equal to StartTime(Y) and EndTime(X) is equal to EndTime(Y). This is illustrated in Figure 22.

Constraint: - StartTime(X) ≠ StartTime(Y); EndTime(X) = EndTime(Y);

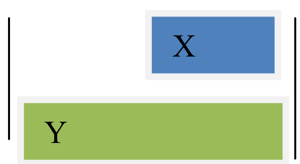


Figure 22: X finishes Y.

<u>X finished-by Y:</u> The relation "finished-by" is used between two events X & Y, when the event X is finished by the event Y and satisfies the constraint that StartTime(X) is not equal to StartTime(Y) and EndTime(X) is equal to EndTime(X). This is illustrated in Figure 23.

Constraint: - StartTime(X) ≠ StartTime(Y); EndTime(X) = EndTime(Y)

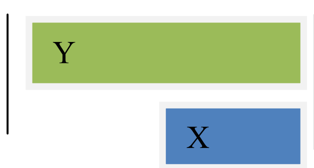


Figure 23: X finished-by Y.

X equals Y: The relation "equal" is used between two events X & Y, when the X occurs at the same time or equally with the event Y and satisfies the constraint that Start Time(X) is equal to StartTime(Y) and EndTime(Y) is equal to EndTime(X). This is illustrated in Figure 24.

Constraint: - StartTime(X) = StartTime(Y); EndTime(X) = EndTime(Y).

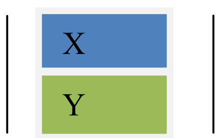


Figure 24: X equals Y.

The thirteen temporal relations are well identified and defined by the boundary conditions as stated. These conditions or constraints are used in the algorithm for identifying temporal intervals.

A question may arise as to why Allen's temporal relations should be used for generating temporal intervals. The temporal relations defined by Allen form our representation of temporal intervals, which when used with constraints become a powerful method of expressing expected temporal orderings between events in a smart environment. In addition, they have an easy naming convention, making it easier to recognize, interpret and use the temporal relations that are identified. There are existing projects which employ sequential information to predict activities [12], and other methods for identifying suspicious states in a smart environment have been explored [14]. The current approach is unique and would incorporate time based relations for analyzing anomalies which is not present in other techniques. But the future work could include fusion these techniques for better analysis. The pseudo code for TempAl's temporal analyzer tool is described in Algorithm 1.

Algorithm 1 Temporal Interval Analyzer

```
Input: data timestamp, event name and state
  Repeat
    While [Event && Event + i found in same day window]
        Find paired "ON" or "OFF" events in data to
            If found save the event name and start and end time of Interval &
              Increment i, Goto next event.
        End while loop until end of events in the input
    While [Interval && other Intervals found in same day window in earlier formed
              Interval dataset]
        Find temporal range, satisfied by constraint and
         Identify relation type between event pair from
           possible relation types (see Table 1).
        Record relation type and related data.
        Increment Pointer
 Loop Until end of input.
```

We extend these methods to incorporate valuable information about the interval of time each event spans. While other methods treat each event as a separate entity (including, for example, turning on a lamp and later turning off the same lamp), our interval-based analysis considers these two events as members of one interval. Each interval is expressed in terms of start time and end time values. As a result, temporal relationships between such intervals can be identified and used to perform critical anomaly detection. We see that there are a few limitations to this process of interval formation, it does not include any intermediate states if any and uses just the ON and OFF states to form interval relations. Some of the future work incorporates intermediate states too. We can find some additional information on this in the future work section too.

Benefits of Temporal Relations

Temporal relations are beneficial in many ways. Reasoning about these relationships aids the processes of reminder assistance, anomaly detection, and temporal need analysis. In event prediction can be improved by incorporating temporal relation information. The benefits are illustrated in Figure 25, with examples describing scenarios where temporal relations can be applied and are most beneficial.

They aid prediction, where given a description of a scenario, which includes actions and events related by temporal relations, we could predict what event will happen next. Temporal relations can also aid planning, where given a description of the world and a desired goal; we can find the course of action that will most likely need to be taken to achieve that goal.

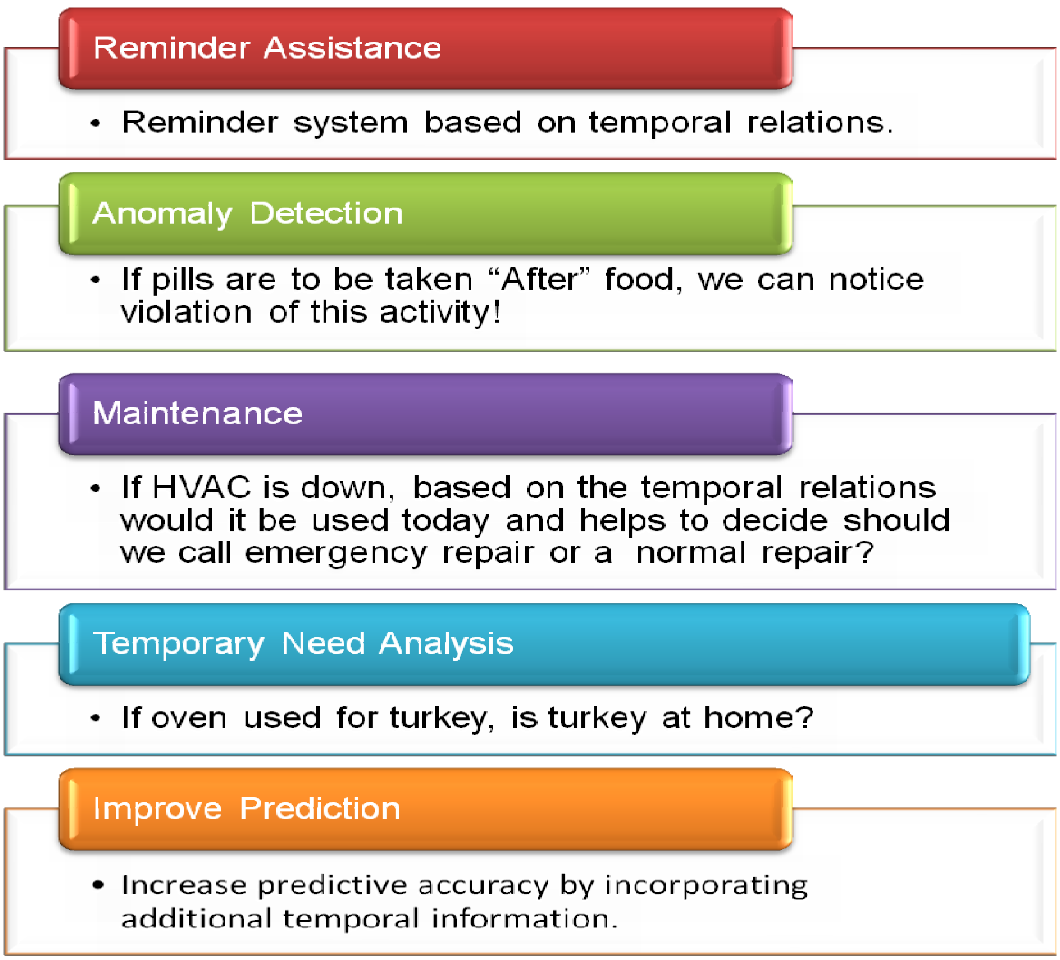


Figure 25. The benefits of temporal relations with examples. Benefits include reminder assistance, anomaly detection, maintenance, temporary need analysis, and improvement of event prediction.

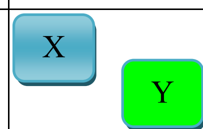
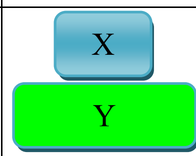
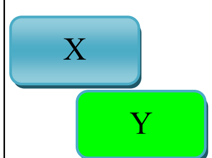
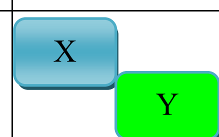
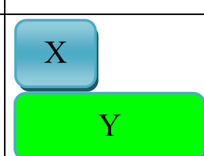
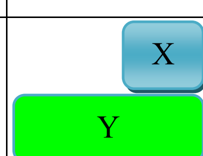
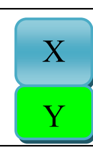
Anomaly Detection Using Temporal Relations

The possibility of designing intelligent systems that can detect when an undesirable situation may arise would make smart homes an environment which can become more supportive of independence and more private than a traditional assisted care facility or a hospital. We define a smart environment as one that collects data about the residents and the environment in order to adapt the environment to the residents and meet the goals of safety, security, cost effectiveness, and comfort. In an environment that is equipped with sensors to detect motion, temperature, and other conditions, sensed events can be captured and associated with a time stamp. These time intervals offer additional information about the relationships between timings of activities that improves the performance of health monitoring tasks such as anomaly detection.

One of the major objectives 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-likely events. To validate the approach, we test the algorithm on synthetic data as well as real data collected from a smart environment. We discuss the implications of this work for health monitoring and assistance.

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 a resident in a smart home and the resident has not opened the refrigerator all day as they normally do, this should be reported to the resident 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.

Table 2. Nine Temporal relations representation which aid anomaly detection (Before, contains, overlaps, meets, starts, started-by, finishes, finished-by, equals) [51].

Temporal relations	**Visualization**	**Interval constraints**
X **Before** Y		Start(X)<Start(Y); End(X)<Start(Y)
Y **Contains** X		Start(X)>Start(Y); End(X)<End(Y)
X **Overlaps** Y		Start(X)<Start(Y); Start(Y)<End(X); End(X)<End(Y)
X **Meets** Y		Start(Y) = End(X)
X **Starts** Y Y **Started-by** X		Start(X)=Start(Y); End(X)≠End(Y)
X **Finishes** Y Y **Finished-by** X		Start(X)≠start(Y); End(X) = End (Y)
X **Equals** Y		Start(X)=Start(Y); End(X)=End(Y)

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 [52] [53]. 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. The Psuedocode for the algorithm is given in algorithm 2.

Algorithm 2 Psuedocode for Apriori Algorithm.

C_k: Candidate itemset of size k
L_k : frequent itemset of size k
L_1 = {frequent items};
For (k = 1; L_k !=∅; k++) do begin
    C_{k+1} = candidates generated from L_k;
    For each day t in datasets do
        increment the count of all candidates in C_{k+1} that are contained in t
    L_{k+1} = candidates in C_{k+1} with min_support
    End
Return $\cup_k L_k$;

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.

The temporal relations that are useful for anomaly detection are the before, contains, overlaps, meets, starts, started-by, finishes, finished-by, and equals relations. 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 Equation (1). Note that the equation 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.

$$P(C|B) = |\text{Before}(B,C)| + |\text{Contains}(B,C)| + |\text{Overlaps}(B,C)| + |\text{Meets}(B,C)| + |\text{Starts}(B,C)| + |\text{StartedBy}(B,C)| + |\text{Finishes}(B,C)| + |\text{FinishedBy}(B,C)| + |\text{Equals}(B,C)| / |B| \tag{1}$$

In Equation 1, we compute the probability of the occurrence of C given that B occurred using the temporal relations frequency shared between the two events C and B. 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:

$$P(C|A \cup B) = P(C \cap (A \cup B)) / P(A \cup B) = P(C \cap A) \cup P(Z \cap Y) / P(A) + P(B) - P(A \cap B)$$
$$= P(C|A).P(A) + P(C|B)\,P(C) / P(A) + P(B) - P(A \cap B) \quad (2)$$

In Equation 2 we look at the calculation of the evidence of occurrence of C (here C is the most recently occurred event) when A and B are both frequent events and both have occurred, the "∪" symbol is the union symbol which is similarly used here, but it can be termed to represent the occurrence of A, B or A&B with the occurrence of C, (where C is currently occurred event for which we are working to determine whether it is a anomaly or not) and the "∩" in "A∩B" represents the intersection operator or occurrence of the A and B (there occurrences together), we see that when we find the occurrences of A and B they might have some temporal relations in common where A involves B and B involves A and these can be removed so that there are no repetitive or additional counts involved. We also need to note that P(A&B|C) is a different causal approach and would be looked upon as given C occurred whether A and B are anomalies, this is a different way of approach for anomaly detection process. Our approach looks at "C" is the most current occurred event and finds evidence whether C is an anomaly or not. This would aid various decision making processes and later when fused with a decision maker, if an anomaly is found then the decision maker calls for the next highly predicted event and uses that for prediction or task automation rather than the current prediction which was found to be a anomaly. The role of C in this equation is that it is the most recent (the current) event which occurred or is predicted, we try to evaluate whether it is an anomaly or not. Using Equation 2 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 also calculate the likelihood that an event C *does not* occur as $P(\neg C) = 1 - P(C)$, or the inverse of the probability that event C does occur. Finally, we calculate the anomaly value of event C using Equation 3.

$$\text{Anomaly}_C = 1 - P\,(C) \tag{3}$$

Notice that if the event has an anomaly probability approaching 1 and the event occurred, this is considered an anomaly. We should note that if the probability is close to 0 and the event does not occur then it is also considered anomaly. However, the current process only calculates the anomaly of the events which do occur and thus the calculation of the anomaly of the events which do not occur is considered to be extension to this work and is suggested for future work. The point at which these anomalies are considered surprising enough to be reported is based somewhat on the data itself. 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. Events are reported as anomalies (or, conversely, the absence of an event) if it does occur and its anomaly value is greater than the mean probability + 2 * standard deviations of the anomaly probability population which is observed. Two standard deviations away from the mean account for roughly 95 percent, so any value which falls out of this population would be reported as an anomaly. The synthetic and real datasets are processed for anomaly detection and the observations are reported as results in Chapter five.

Enhancing Prediction Using Temporal Relations

Earlier work in the MavHome project researched methods of performing sequential prediction of smart home events [8]. We extend this work to incorporate valuable information about the time interval each event spans. While other methods treat each event as a separate entity (including, for instance, turning on a lamp and later turning off the same lamp), our interval-based analysis considers these two events as members of one interval. Each interval is expressed in terms of start time and end time values. We see that the interval formulation has a few limitations, which includes limiting the interval formation to include pairs of ON and OFF states only. In this study the intermediate states are currently discarded, but can be included in future work. Another limitation is related to how the states are formed. We preprocess the event history to look for pairs of ON and OFF states which form an event interval. If an end of the event is not found, TempAl considers the end of the day as when the event ends and considers the corresponding delay to be the length of the interval. To introduce the topic of prediction using temporal intervals, let us consider a typical activity in a smart environment, which includes a television, a fan and a lamp. Now we apply Allen's logic to establish the temporal relations among the activities which occurred. These activities can be represented as television "contains" fan and "meets" lamp. We can also represent these relationships as television "meets" lamp and fan "before" lamp. We can see that based on the occurrence of these actions we can use these for prediction. The nine out of thirteen possible temporal relations which aid the process of prediction are shown in Table 3. Consider two general events X and Y; we use these variables to represent relations in the table. In Table 3, the interval constraints compare the start time (StartTime) and end time (EndTime) of the activities, X and Y. In the table we can note that the visualization includes a line based representation of the time and only the end points of this line representation are to be considered as the start and end times of the event occurrence.

Table 3. Prediction aiding Temporal relations (After, During, Overlapped-By, Met-By, Starts, Started-By, Finishes, Finished-By, and Equals).

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

We consider two alternative approaches to enhancing prediction with temporal relation information. In the first approach, we extract association rules which can be used for prediction. Using the second approach, we integrate temporal information into the ALZ prediction algorithm [46]. For the first approach rules which are formed with association mining are actually used to predict the events. This can be done as follows: we run the ALZ algorithm then see if its output is found in any of the top association rules. Here we input that the most current occurred event and then check if it is an antecedent of any rule. If the observed event matches the antecedent of a rule, then the corresponding rule's consequent is output as TempAl's prediction. Association rule mining identifies correlations between events that occur frequently and occur often together. Association rule mining is used to form strong rules based on a set confidence and support. Two events are associated by each rule and they are of the form (Event 1, Temporal Relations) → Event 2. If the predicted event is found on the left hand side of one of these rules, then we calculate the evidence of occurrence of event appearing on the right hand side of the rule. We need to note that there can be more than one event on the consequent of the rule. In this case, we see that both consequent events are considered as the possible next events. We compute the evidence for all of the consequent events in the same order and if satisfied they are predicted in the same order present in the rule. We have a discussion on the issue of deciding on how to handle rules with similar confidence in the future work section. We also have some issues with the precision of the rules. The question is whether we should consider all of the rules or instead we should be selective. This issue is also discussed in the future work session. If this evidence is found to be greater than the sum of the mean evidence probability plus two times the standard deviation of the probability population, then we output the corresponding event as the predicted event. The evidence is calculated using the temporal rules and is discussed below in an elaborate manner.

The second approach we evaluate using TempAl is to include the temporal probability into the ALZ prediction calculations. We need to note that the window size of one day is considered for event interval formation and for calculating the probabilities of event occurrences, but when it comes to finding the association rules we see that we need to generate them over a moving window of four weeks would make the whole process more efficient and would adapt to the changing activities of the inhabitant. Now let us look at examples to illustrate both temporal-based prediction approaches. For the Association Rule Mining based approach let us consider a example where we have a strong rule identified which states the occurrence of a television event frequently coincides with the occurrence of lamp event. We see that when the television event is the most recent observed event we can calculate the evidence of a lamp event. If this condition is satisfied, the lamp event is output as the next prediction, In contrast, using the second approach we calculate the probability of the most likely event to occur based on historical information both for resident events as well as the temporal relationships between frequent events.

Association Rule Mining Approach for Enhancing Prediction

This model acts as a simple rule-based processing model. We mine frequent event sequences and use them to enhance prediction. In this step, we identify the best association rules using Weka [54] which can be used for prediction. The Weka implementation of an Apriori-type algorithm is used, which iteratively reduces the minimum support until it finds the required number of rules within a given minimum confidence. The final step involves calculating the evidence of the event occurrence, which can be used for calculating the prediction. This step is designed to detect whether the particular event satisfies the temporal relations that can be used for prediction. This method of prediction is based entirely on normative behavior as observed in the past and on which basis a strong rule is identified. As a result, the likelihood of prediction increases when there are strong repetitions of resident patterns over time. This method is a probability-based model which involves calculating the evidence supporting the currently-occurring event with respect to the previously-occurred events.

Association rule learners are used to discover elements that co-occur frequently within a data set [52] [53] [54] consisting of multiple independent selections of elements (such as purchasing transactions), and to discover rules, such as implication or correlation, which relate co-occurring elements. Apriori is a classic algorithm for learning association rules and we use the Weka implementation of Apriori for this experiment. As is common in association rule mining, given a set of transactions (for instance, sets of retail transactions each listing resident items purchased), the algorithm attempts to find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found. Now let us look at the various steps involved in this experiment process. Consider a simple instance where we have two events Q and R which also end up being members of a frequent itemset.

Step A: Learn temporal relations from the observed event history by analyzing the resident's events. Use this temporal relations dataset and mine for strong association rules (strong here refers to the best rule found for a given a confidence and support) which can be used for prediction.

Step B: Determine the most recent event. We use ALZ to do this for our experiments, by retrieving its most recent prediction and letting this event represent the recent event.

Step C: Once we have the set of events the the most recent event Q is associated with, we calculate the evidence supporting the occurrence of those events. Let us assume the event associated with Q is R. The formula to calculate the evidence using temporal relations is given by Equation (4).

$$P(R|Q) = |After(Q,R)| + |During(Q,R)| + |OverlappedBy(Q,R)| + |MetBy(Q,R)| + |Starts(Q,R)| + |StartedBy(Q,R)| + |Finishes(Q,R)| + |FinishedBy(Q,R)| + |Equals(Q,R)|/|Q| \quad (4)$$

$$Evidence_X = P(X) \quad (5)$$

Note that equation is based on the observed frequency of the temporal relations, specifically those that influence the occurrence of event X. The previous discussion showed how to calculate the likelihood of event X given the occurrence of one other event Y. Notice that if the event has a probability approaching 1, this is considered most likely to occur. We calculate the mean and standard deviation of event frequencies over the set of events in the resident's action history. Events (or, conversely, the absence of an event) are reported as predicted if it does (does not) occur and its prediction value is greater than the mean probability + 2 * standard deviation (or less than the mean – 2 * standard deviation). We also need to note that, even if multiple events are identified as strongly associated with the most recent event, we calculate their probabilities individually with respect to Q and then calculate their respective evidences. And the rest of the process is performed as before.

Step D: In this final step we see if the computed probability is greater than or equal to the set bound (sum of mean and two * standard deviation). If so, we use this event as the next predicted event and check for accuracy by comparing it with next event that occurred in the test data. Next, the test set event is incremented and this event is inserted into the existing ALZ based prediction tree.

The pseudo code for this first approach is given in Algorithm 3, which is the simplest representation of the logic to be used for the experimentation of using association rule mining approach for incorporating temporal information for enhancing prediction.

Algorithm 3 Psuedocode Temporal Rules Enhanced prediction.

Input: Output of ALZ Predictor a, Best Rules r, Temporal Dataset

Repeat

If a! = *null*

Repeat

Set r1 to the first event in the relation rule

If (r[i].relationoccur ==a) **Then** Read r[i].relationpredict, if any

Calculate evidence of r[i].relationpredict using temporal dataset.

If evidence > (Mean + 2 Std. Dev.)

Then predict;

Else

Continue;

End if.

Until end of rules.

End if.

Loop until End of Input.

To evaluate the benefits of this approach, we compare the performance of ALZ with and without use of the association rules. We notice that many situations demand that the prediction algorithm be capable of analyzing information and delivering in real time. We currently plan to run real time analysis over large sets of data in the near future. The experimental results are displayed in chapter five.

Enhancing Prediction by Adding Temporal Relations-based Probabilities to ALZ

The existing prediction model is a sequential predictor called ALZ [12]. In an event-driven system there is a need to predict the next action in order to provide a clear understanding of the current state. The system will need to make this prediction based only on previously-acquired knowledge. Currently, we use the Active LeZi (ALZ) algorithm [12] with the Transaction Directed Acyclic Graph (TDAG) [48] algorithm to meet our prediction requirements. ALZ is also inherently an online algorithm, since it is based on the incremental LZ78 data compression algorithm [48]. ALZ incorporates a sliding window. The pseudocode of the basic algorithms which are a part of the ALZ is given in Algorithms 4 and 5.

Algorithm 4 Psuedocode for LZ78 [48]

```
Loop
        Wait for next symbol v
        If ((w.v) in dictionary):
                w = w.v
        Else
                Add (w.v) to dictionary
        w = null
        Increment frequency for every
        Possible prefix of phrase
Forever
```

Algorithm 5 Psuedocode for ALZ [48]

```
Initialize Max_LZ_length = 0
Loop
```

```
        Wait for next symbol v
        If ((w.v) in dictionary):
                w: = w.v
        Else
                Add (w.v) to dictionary
        Update Max_LZ_length if necessary
        w: = null
        Add v to window
        If (length (window) > Max_LZ_length)
                Delete window [0]
        Update frequencies of all possible
        Contexts within window that includes v
Forever
```

In order to predict the next event of the sequence for which ALZ has built a model, we calculate the probability of each state occurring in the sequence, and predict the one with the highest probability as the most likely next action. In order to achieve better convergence rates to optimal predictability, the predictor must "lock on" to the minimum possible set of states that is representative of the sequence. For sequential prediction, it has been shown that this is possible by using a "mixture" of all possible order models (phrase sizes) to assign the next symbol to its probability estimate. To consider different orders of models, we turn to the Prediction by Partial Match (PPM) family of predictors. This has been used to great effect by Bhattacharya and Das [55] for a predictive framework based on LZ78, but their method only concentrates on the probability of the next symbol appearing in the LZ phrase, as opposed to the next symbol in the sequence.

Consider the sequence x_Z= aaababbbbbaabccddcbaaaa. An LZ78 parsing of this string would yield the phrases as displayed in Figure 26. As described above, this algorithm maintains statistics for all contexts seen within the phrases w_i. For example, the context a occurs 5 times (at the beginning of the phrases a, aa, ab, abc, aaa) and the context bb is seen 2 times (phrases bb and bba) [48].

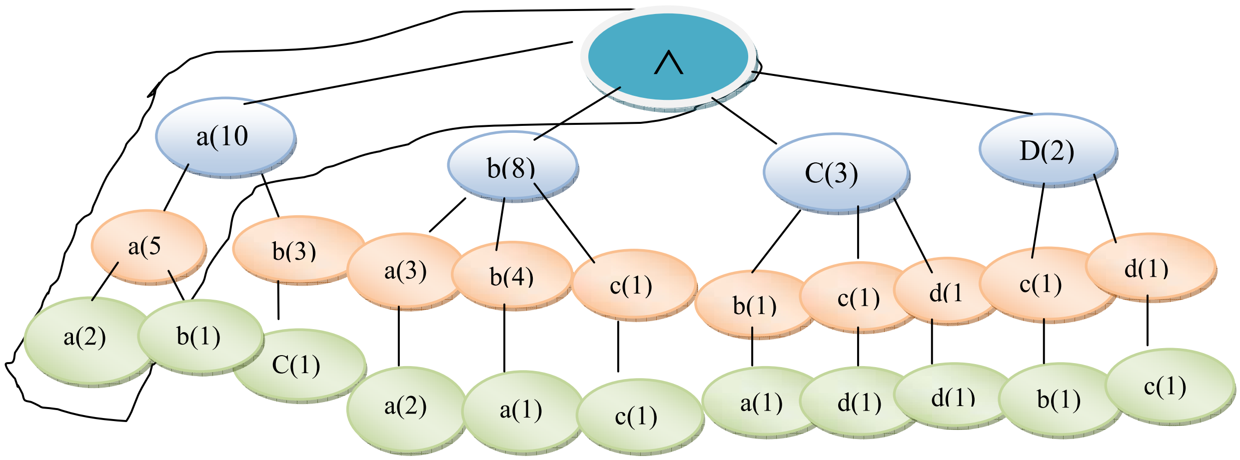


Figure 26. Trie formed by the ALZ parsing of the sequence *aaababbbbbaabccddcbaaaa*. The selected path acts as the phrase for the context of each probability computation [48].

As the Active LeZi algorithm parses the sequence, larger and larger phrases accumulate in the dictionary. As a result, the algorithm gathers the predictability of higher and higher order Markov models, eventually attaining the predictability of the universal model. Let us now look at how the probability is computed. Suppose we need to compute the probability that the next symbol is an *a*. From Figure 3, we see that an *a* occurs two out of the five times that the context *aa* appears, the other cases producing two null outcomes and one *b* outcome. Therefore the probability of encountering an *a* at the context *aa* is 2/5, and we now "escape" to the order-1 context (i.e., switch to the model with the next smaller order) with probability 2/5. This corresponds to the probability that the outcome is null, which forms the context for the next lower length phrase. At the order-1 context, we see an *a* five out of the ten times that we see the *a* context, and of the remaining cases, we see two null outcomes. Therefore we predict the *a* at the order-1(orange color in figure 3) context with probability 5/10, and escape to the order-0 model with probability 2/10. Using the order 0 (blue color in figure 3) model, we see a ten times out of the 23 symbols processed so far, and we therefore predict *a* with probability 10/23 at the null context. As a consequence, the *blended* probability of seeing an *a* as the next symbol is computed as [48]:

$$\frac{2}{5}+\frac{2}{5}\left\{\frac{5}{10}+\frac{2}{10}\left(\frac{10}{23}\right)\right\}.$$

Now in this current technique we enhance this propability by incorporating the temporal probability at higher order level. Here we just add temporal information to the sequential information as at the higher order we should note that the temporal probability holds more information than sequential probability. The resulting probability can be the sum computed of both these probabilities as illustrated in Equation 6. Earlier we looked at an instance of sequential probability being calculated. Now we look at how the temporal probability be calculated. We enhance existing ALZ prediction by incorporating the temporal probability with the sequential probability at each higher order-level of the phrase (for instance the phrase is BC) as follows:

$\text{Prediction}_c = P(C|B) = P(C|B)_{\text{SEQ: Order-0}} + P(C|B)_{\text{TEMPORAL: Order 1-n}}$ (at each order in phrase) $+ P(C|B)_{\text{SEQ: n-}\infty}$ (6)

Probabilities at the 0 context size are drawn from the ALZ trie. Similarly, probabilities for context sizes greater than 1 are calculated from the ALZ trie. The probability for the size 1 context, on the other hand, uses the TempAl calculation. The TempAl formula uses all of the information available to ALZ plus the temporal relationship information. The reason we fuse these two probabilities as we note that at the higher order we see that temporal probability would include more information compared to the sequential probability. The sequential probability will only include the “before” temporal relation in the calculation but the temporal probability will include information from all thirteen relationship types.

Now consider the case where we want to combine evidence from multiple events, in the phrase which belongs to the current context window, that have a temporal relationship with X. 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 X occurring. From Equation 6 we can calculate the evidence.

Now we calculate the evidence of B as:

$$P(B|A) = |After(B,A)| + |During(B,A)| + |OverlappedBy(B,A)| + |MetBy(B,A)| + |Starts(B,A)| + |StartedBy(B,A)| + |Finishes(B,A)| + |FinishedBy(B,A)| + |Equals(B,A)| / |A| \quad (7)$$

We also use this information to calculate the evidence of the most recent occurred event.

Similarly when we have the events occurred as follows: **A B X**

Now the evidence of B is calculated as follows:

$$P(X|A \cup B) = P(X \cap (A \cup B)) / P(A \cup B) = P(X \cap A) \cup P(X \cap B) / P(A) + P(B) - P(A \cap B)$$

[Distributive Rule]

$$= P(X|A).P(A) + P(X|B).P(B) / P(A) + P(B) - P(A \cap B)$$

[Multiplication Rule]

(8)

We can use the previous calculated evidence for calculating newer evidence, based on Equation (7). We use the distributive and multiplicative rules to arrive at the final formula shown in Equation 8 which includes the previously- computed evidences of occurred events. This evidence calculation helps to compute the temporal probability of the event to occur. This evidence is used for the temporal probability calculation which is incorporated into the Alz probability estimation. Now we finally calculate the temporal probability using Equation (9).

$$\text{Temporal Prediction}_X = P(X) \tag{9}$$

Here we want to predict the event with greatest probability. We see that we have combined the temporal information with the existing sequential predictor enriching it to make better predictions. Similar to the above explanation of probability calculation we add temporal propability for that particular activity. Experimental results obtained with both of these approaches are summarized in Chapter five.

Data Visualization

In many application domains temporal data queries can help analysts understand data and reason about it over time and space. The understanding is further enhanced when the temporal relationships can be visualized well. However it still would be a difficult challenge to describe and formalize time using physical symbols, characteristics and properties. In this chapter, we present an interface which helps us visualize activities using time intervals identified in smart home datasets [66]. We can also use this tool as a media for looking for patterns of interest and also act as a user-friendly data visualization tool. The end goal of this tool is to enable the resident to visualize interesting patterns over time and help his improve his lifestyle at home by identifying and changing activities which hinder his growth and health and identify activities which help him to have an improved lifestyle.

As we use temporal related datasets, and the window size is spanned across a single day, we mine using the longest common subsequence technique [56] [60] with maximal consecutive events, where we find a long subsequence or a common subsequence of all sequences in a set of sequences when we compare two days at a time. In our implementation, we look for maximal consecutive sequences within a two-day period. The reason for performing a maximal consecutive look up is that a three commonly-shared subsequence is of significant interest compared to one sequential subsequence. For instance, consider two strings X and Y, where X = "**AA** BB PP **BB** ZZ **CC** KK **DD** UU VV **EE** RR" and Y = "**AA** **BB** **CC** **DD** EE FF GG". Using this technique, finding that AA BB CC DD EE occurs more often is of more interest compared to finding a single occurrence of the sequence AA BB CC DD. This approach is a LCS sequence mining [62] method to find interesting temporal sequences, i.e. finding interesting temporal patterns among everyday activities in a smart environment.

This tool was developed using C#.NET with Visual studio 2005 IDE and uses VARCHART Xgantt plotting library (a 3rd party library) for visualization in the form of Gantt charts which help represent the activity intervals over a time period [58] [59].

Here the pattern search tool identifies common patterns for two days at a time for comparison. Later we analyze the attributes of interestingness over the longest common subsequences identified in the experimental datasets. The longest common subsequence problem is NP-Hard for a general case of arbitrarily long input sequences and the problem is solvable in polynomial time using dynamic programming [57]. The reason we are interested in longest length and not the shortest one is because the shortest ones just identify the simple temporal relations which can be identified using other frequent mining techniques but the interesting part is the longest non-sequential patterns because they are unique for most of the days based on the resident. The future work for this tool involves evaluation by the residents, where the resident uses this tool to look for patterns and later make changes accordingly and rate whether this particular tool did help them improve their lifestyle. This is an additional tool and uses temporal intervals for visualization and also has a pattern search tool for finding common temporal relations.

Summary

In this chapter we had a detailed look at the techniques being used by *"TempAl"* for the process of anomaly detection and enhancing prediction. We also looked at a visualization tool which can be interfaced with a temporal pattern search tool to look for interesting patterns. In the next chapter we report the findings of experiments that analyze the effectiveness of these algorithms.

CHAPTER FIVE
EXPERIMENTATION FINDING AND DISCUSSIONS

In this chapter we report the results of experiments we conducted to evaluate the effectiveness of smart home algorithms enhanced with temporal relation information. We report the temporal relations that are discovered in real and synthetic smart home datasets, and report the results of anomaly detection and prediction applied to these datasets with the corresponding temporal relations.

Temporal Relation Formation

Sensor data from a smart environment can be represented and mined as sequences or as time series data. A sequence is an ordered set of events, frequently represented by a series of nominal symbols [61]. All the sequences are ordered on a time scale and occur sequentially one after another. However, for some applications it is not only important to have a sequence of these events, but also a time interval as when these events occur. This is particularly true for smart homes. A time series is a sequence of continuous real-value elements [17] [61]. This kind of data is obtained from sensors which continuously monitor parameters such as motion, device activity, pressure, temperature, brightness, and so forth. Each time stamped data point is characterized by specific properties. Table 4 describes the number of events, number of events, and number of temporal intervals that were identified in the synthetic and real datasets used for our experiments.

In Table 5, we illustrate a sample of raw data collected from the sensor and include the data as how it looks after it is processed and temporal intervals are identified. Figure 27 shows the various stages involved in the conversion of the raw data to a temporal relations dataset.

Table 4. Parameter settings for experimentation.

Datasets	Parameter Setting			
	No of Days	No of different Events[Devices]	No of Intervals Identified	Size of Data
Synthetic	60	8	1729	106KB
Real	60	17	1623	104KB

Table 5. Sample display of sensor data across various stages of temporal relation formation. (Note: Most experiment processes are performed on the temporal datasets).

Raw Sensor Data		
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

Identify Time Intervals			
Date	Sensor ID	Start Time	End time
03/02/2003	G11	01:44:00	01:48:00
03/02/2003	G19	02:57:00	01:48:00
03/02/2003	G13	04:06:00	01:48:00
03/02/2003	G19	04:43:00	01:48:00
03/02/2003	H9	06:04:00	06:05:00
03/03/2003	P1	10:55:00	17:28:00
03/03/2003	E16	11:18:00	11:34:00
03/03/2003	G12	11:23:00	11:24:00

Temporal Relations

Date	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

Figure 27. The steps involved in the processing of temporal relations formulations in datasets.

The first step of the experiment is to process the raw data to find the temporal intervals. This is done using a simple tool which takes the timestamp of the event that occurred and based on the state (ON or OFF) forms the intervals. Later this data is passed through the temporal analyzer tool which identifies the temporal intervals based on the constraints formulated. This process is illustrated in Figure 25.

Anomaly detection

We validate our algorithms by applying them to our real and synthetic datasets. We train the model based on 59 days of data and test the model on a single day of observed activities. We use the training set to form the frequent item sets to identify frequent activities which happen in the smart home.

Next, we identify temporal relations in all of the datasets that are shared between events which occur in a smart home. The temporal relations formed in these data sets show some interesting patterns and indicate relations that are of interest. Table 6 and Table 7 summarize the characteristics of the datasets we used for the experiments.

Next, we perform frequent itemset mining and identify the most frequent activities in the training dataset. Then we read these temporal relations into our anomaly detection tool which calculates evidence for each possible event and outputs anomalies that are detected in the test set data. After manually inspecting the data, we report the number of true and false anomalies that are reported. Tables 6 and 7 display results from the synthetic and real datasets, respectively. Because anomalies are detected in real time as events are observed, we list anomalies in the order they are detected.

Table 6. Characteristics of the synthetic and real training datasets for anomaly detection.

Datasets[Training]	#Days	#of different events[Devices]	#Identified frequent intervals	Size
Synthetic	59	8	1703	105KB
Real	59	17	1523	103KB

Table 7. Characteristics of the synthetic and real test datasets for anomaly detection.

Datasets [Test]	#Days	# of different events[Devices]	#Identified frequent intervals	Size
Synthetic	1	8	17	2KB
Real	1	17	9	1KB

One of the reasons we use these MavLab datasets is due to the fact that the MavLab environment is full of X10-based devices which monitor resident activity, thus we focus on evaluating these techniques on these datasets. We evaluate the algorithm using the available real and synthetic datasets. Tables 8 and 9 report the observations. Table 8 shows the observed results of the anomaly detection process on the real datasets. Table 9 shows the observed results of the anomaly detection process on the synthetic datasets.

Table 8: Anomaly detection in the test set for the real dataset.

Frequent Events in Chronological Order	Frequent Event	Computed Evidence	Computed Anomaly	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 St. Dev.			0.07	
Anomaly Cut-off Threshold (Mean + 2 * St. Dev)			0.84	

Table 9. Anomaly detection in the test set for the synthetic dataset

Frequent Events in Chronological Order	Frequent Event	Evidence	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	1.0	Yes
16	Fan	0.34	0.66	No
17	Cooker	0.42	0.58	No
Anomaly Mean			0.76	
Anomaly St. Dev.			0.14	
Anomaly Cut-off Threshold (Mean + 2 * St. Dev)			0.99	

Based on a manual inspection of the data we see that the anomaly detection algorithm performed well on synthetic data – all of the expected anomalies were detected and no false positives were reported. In the real data no anomalies are reported.

We should note that synthetic test data has limited number of events and the scenarios are scripted in such a way that events occur weekly and some of them occur very frequently with a difference of couple of hours. We see the event occurrences are randomized and the occurrence can lead to an anomaly as the events could occur based on the time of occurrence and lead to a new set temporal relations rather than a found specific pattern, thus this could make the event in the synthetic data to occur as an anomaly. In the current synthetic test data we see that the occurrence of lamp was noted as an anomaly as in temporal relations history lamp shares weak temporal relations with events that are a part of the test set data and hence an anomaly is reported.

The experiment is consistent with the nature of the data which does not contain anomalous events, and reflects the fact the anomalies should be, and are in fact, rare. We see that the approach is robust and does not report false anomalies in this case. The graph in Figure 28 visualizes the anomaly values for frequent events in the synthetic and real datasets. We notice that the spikes visible in the synthetic datasets are clear indications of anomalies, which is consistent with our expectation for the outcome of this experiment.

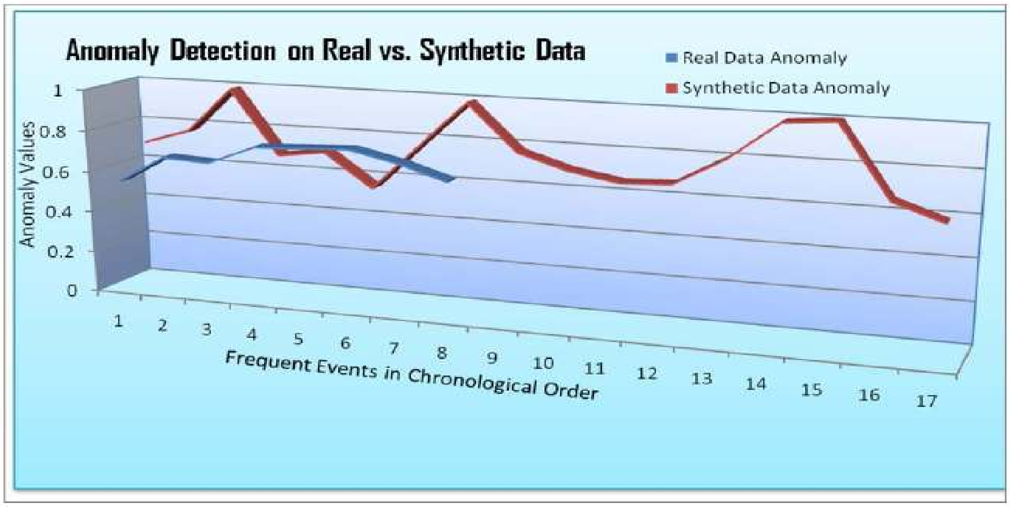


Figure 28. Anomaly detection on test sets of real and synthetic data in 3-D. The anomaly value is plotted for each possible activity as actual events are observed.

Discussion on Anomaly Detection

The experimental results [51] on synthetic data provide evidence that our algorithm is capable of identifying anomalous events based on temporal relationship information. The results applied to real data brought insights to the activities that were being performed in the MavLab setting. In both cases these types of surprising behaviors should 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 may have available.

An extended application of anomaly detection is its use for reminder assistance. If the resident queries the algorithm for the next routine activity, the activity or activities with the greatest probability will 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 is an example of a reminder system that has already been developed using techniques such as dynamic programming and Bayesian learning to remind individuals about their planned Activities for Daily Living. Unlike our approach, Autominder does not base its reminders on a model of behavior that is learned from actual observed events.

Enhancing Prediction

We validate our algorithm by applying it to our real and synthetic datasets. We train the model based on 59 days of data and test the model on one day of activities. We use the training set to form association rules using Weka for the association rule mining based model of prediction and identify temporal relations shared between them. The temporal relations formed in these data sets show some interesting patterns and indicate relations that are of interest. The parameter settings pertaining to the training set data are given in Table 10. The parameter settings pertaining to the test set data are given in Table 11. These datasets are used for both the models of prediction experimentation.

Table 10. Parameters setting for training set for prediction experiment.

Datasets	Parameter Setting			
	No# Days	No # of Different Events	No# Intervals Identified	Size of Data
Synthetic	59	8	1703	105KB
Real	59	17	1523	103KB

Table 11. Parameters setting for test set for prediction experiment.

Datasets	Parameter Setting		
	No # Days	No # of Different Events	Size of Data
Synthetic	1	8	2KB
Real	1	17	1KB

Association Rule Mining Approach for Enhancing Prediction

After the parameters are set and the training and testing data is identified, in the next step, we identify the association rules using Weka, which in-turn can be used for prediction. The Weka implementation of an Apriori-type algorithm is used, which iteratively reduces the minimum support until it finds the required number of rules within a given minimum confidence. Table 12 summarizes the parameters that were set and the number of rules generated with a given specified minimum confidence for the real dataset. Table 13 summarizes the same for the synthetic dataset.

Table 12. Parameter settings and rules generated using Apriori-type algorithm in Weka for real dataset.

Run#	Minimum Support	Minimum Confidence	No of Best Rules Found
1	0.00	0.5	100
2	0.01	0.5	006
3	0.02	0.5	002
4	0.05	0.5	001

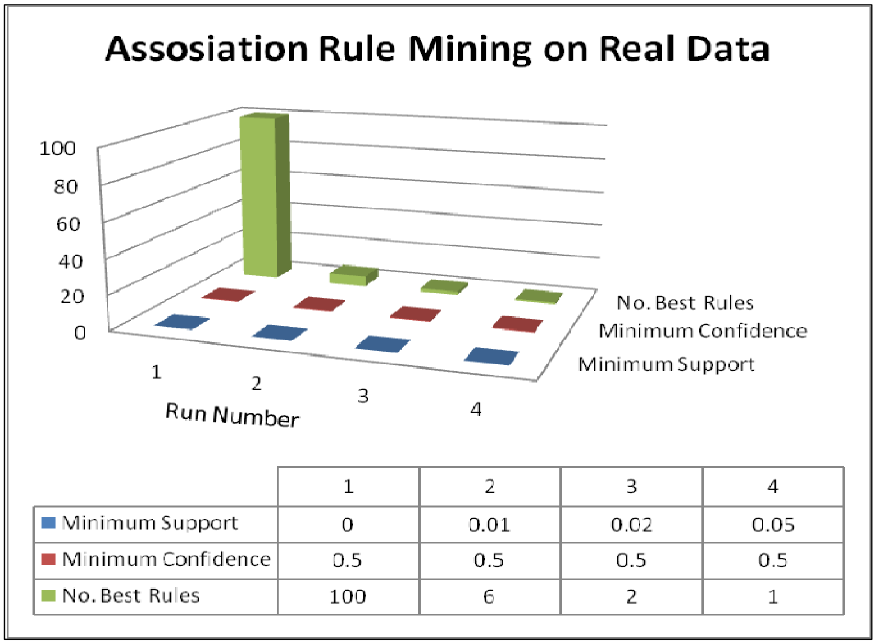


Figure 29. Association rule mining in real datasets.

Table 13. Parameter settings and rules generated using Apriori-type algorithm in Weka for synthetic dataset.

Run #	Minimum Support	Minimum Confidence	No of Best Rules Found
1	0.00	0.5	100
2	0.01	0.5	010
3	0.02	0.5	005
4	0.05	0.5	003

We see that the Figure 29 and Figure 30 represent the observations from tables 5 and 6. We see that they represent various configurations which were used in Weka to find the best rules which can aid the prediction process. We observe here that when there is no minimum support the algorithm generates a large number and as the support is increased we see that the number of rules generated decreases. A sample of generated rules is given in Table 14.

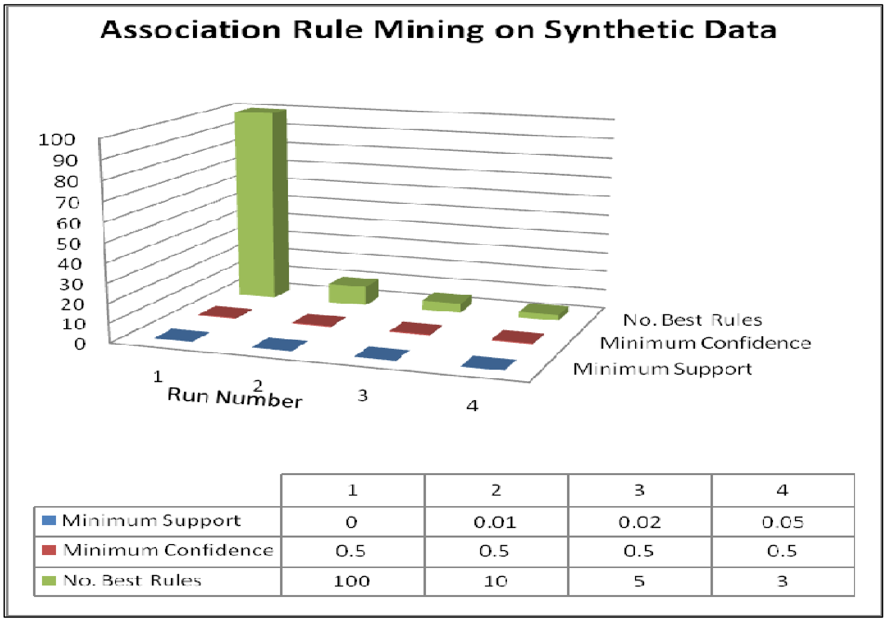


Figure 30. Association rule mining in synthetic datasets.

In Tables 12 and 13, the confidence level above 0.5 and support above 0.05 could not be used, as they could not result in any viable rules, due to the small size of the datasets being used. As we see that the datasets are small, we use the top rules generated with a minimum confidence of 0.5 and a minimum support of 0.01.

Table 14. Display of a sample of best rules generated.

Sample of best rules observed in real datasets: Activity=C11 Relation=CONTAINS 36 ==> Activity=A14 36 Activity=D15 Relation=FINISHES 32 ==> Activity=D9 32 Activity=D15 Relation=FINISHESBY 32 ==> Activity=D9 32 Activity=C14 Relation=DURING 18 ==> Activity=B9 18

The final step would involve calculating the evidence of the event occurrence, which can be used for calculating the prediction on a moving window. This purpose of this step is to detect whether the particular event satisfies the temporal relations which can be used for prediction given in a specified recent history of activities. More discussion on this avenue of research can be found in the future work section. These temporal relations which can be used for prediction are listed in Table 3. Let us look at an example where we have three frequent activities which occur in the order of turning a toaster, table lamp and radio on and off in the morning. We see that the relation exhibited by them can be toaster "before" table lamp "finishes" radio. We need to note that the intervals are formed when a complete cycle of a device from an ON to OFF or an OFF to ON state is pursued within a window of a single day. Now when the toaster and the radio occur without the table lamp, we can note that this is an anomaly in activity and we can use the same relation as when the toaster occurred and table lamp occurred then we can predict that the radio is going to occur in the near future before the table lamp is turned off. This method of prediction is based entirely on normative behavior as observed in the past and a strong rule is identified. As a result, the likelihood of prediction increases when there are strong repetitions of resident patterns over time which are not anomalies. This method is a probability-based model which involves calculating the evidence supporting the currently- occurring activity with respect to the previously-occurred activates.

Finally we enhance ALZ predictor [48] with incorporating temporal relations with the input data and compare the performance with and without these rules. We notice that many situations demand that the prediction algorithm be capable of analyzing information and delivering in real time. We currently plan to run real time analysis over large sets of data in the near future. These rules based system pose a challenge in terms of how do we differentiate rules using a interestingness measure and also would push such rule based systems into the domains of planning and reminder assisting systems.

Table 15. Comparing ALZ based prediction with and without temporal rules in real datasets.

Datasets	Percentage Accuracy	Percentage Error
Real (Without Rules)	55	45
Real (With Rules)	56	44

Table 26. Comparing ALZ based prediction with and without temporal rules in synthetic datasets.

Datasets	Percentage Accuracy	Percentage Error
Synthetic (Without Rules)	64	36
Synthetic (With Rules)	69	31

Tables 15 and 16 present the results of our prediction experiment. We need to note that percentage accuracy is computed as the ratio of the count of number of correct predictions to the total number of predictions. Both percentage accuracy and percentage error are rounded to the nearest unit value. Illustrations of the observed accuracy and error values in the real and synthetic datasets are visualized in Figures 31 and 32, respectively.

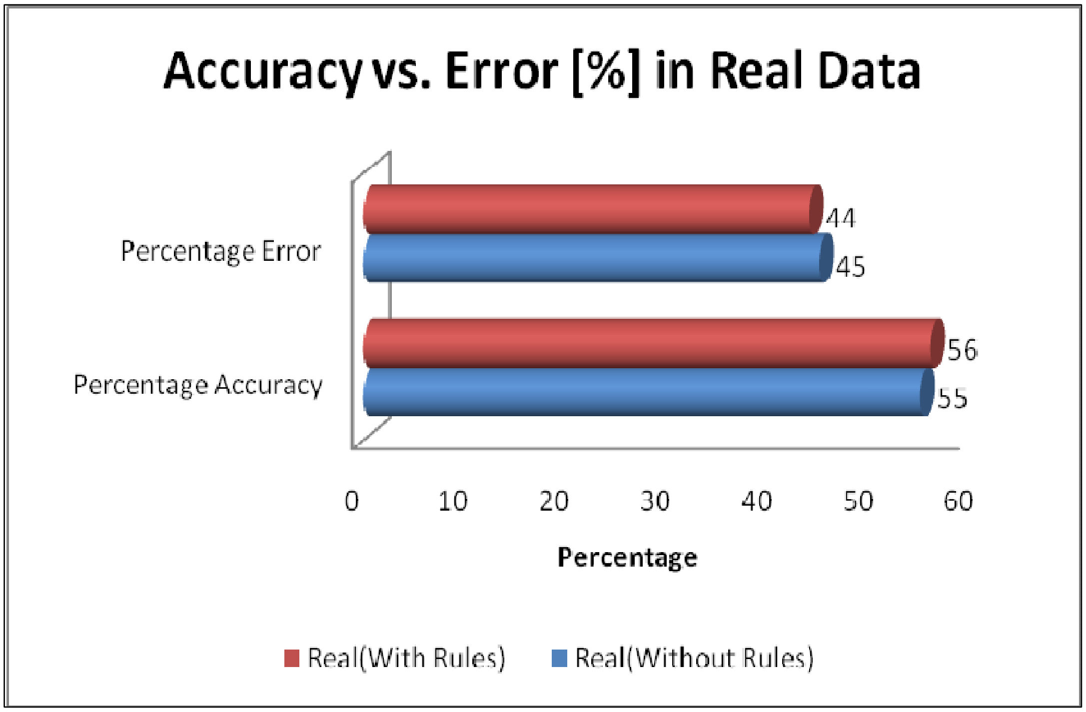


Figure 31. Prediction percentage accuracy vs. percentage error in real datasets using association rule mining.

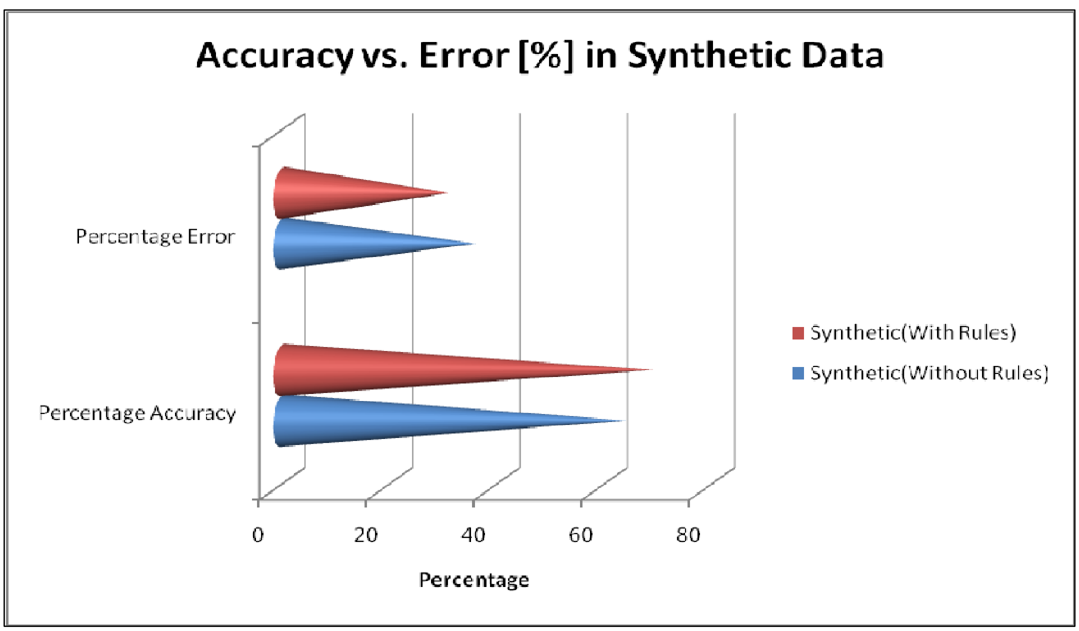


Figure 32. Prediction percentage accuracy vs. percentage error in synthetic datasets using association rule mining.

Discussion on Association Rule Mining Approach for Enhancing Prediction

In this approach [62] we deal with leveraging association rules for prediction, where we see that these rules are used in the form "IF X THEN Y". The consequent part of this rule (Y) can be predicted based on occurrence of X. The main reason for a significant error rate is the smaller amount of data used. As we have larger datasets we see that the performance of the temporal relations enhanced prediction would also improve drastically over time. Another cause of the error rate and means to better performance is making the right trade-off while choosing the support and confidence levels for the discovery of these association rules. The refinement of association rules by including an interestingness factor would make the rules more precise and might push towards better prediction accuracy. Table 15 and 16 summarize the observed accuracy of the prediction performance on real and synthetic datasets. We see that there was around 1% prediction performance improvement in the real data and around 7% improvement in the synthetic data. This indicates an improvement of event prediction in a single day of the resident in smart environment.

The main reason for the error rate is the small amount of training data. With larger datasets we would expect to see that the performance of the temporal relations enhanced prediction would also improve drastically over time. Overall we see a unique application of temporal relations based mining being applied. The basic idea of the association rule based prediction is to develop a rule based system which enhances performance of the event predictor.

A possible next step for this approach would be to evaluate these association rules for interestingness which involves applying spatial techniques along with temporal analysis to determine which of the identified rules are of interest and would help prioritize the generated rules that have equal confidence and support values.

Enhancing Prediction by Adding Temporal Relations based Probability to ALZ

In this experiment we leverage the existing prediction using temporal information as an additional source to evaluate the next occurring event and thus aid prediction. For this approach we validate our algorithm by applying it to our real and synthetic datasets. We train the model based on 59 days of data and test the model on one day of activities. The temporal relations formed in these data sets show some interesting patterns and indicate relations that are of interest. The parameter settings pertaining to the dataset are given in Table 17.

Table 17. Dataset descriptions of training and test set used for experimentation.

Datasets	No # Days	Total No # events
Real (Train)	59	750
Real(Test)	1	40
Synthetic (Train)	59	13900
Synthetic (Test)	1	1500
Cross Validation (Real)	60	834
Cross Validation (Syn.)	60	15000

Table 18. Comparing accuracy of prediction techniques using TempAl on real datasets

Dataset (Learning Algorithm)	Train	Test	Correct	Prediction Accuracy (%)	Prediction Error (%)
Real (Alz)	100	1	0	0%	100%
Real (Alz+Tempal)	100	1	1	100%	0%
Real (Alz)	100	10	6	60%	40%

Real (Alz+Tempal)	100	10	6	60%	40%
Real (Alz)	750	40	29	72.50%	27.50%
Real (Alz+Tempal)	750	40	29	72.50%	27.50%
Cross Validation (Alz)	787	83	48	57.96%	42.04%
Cross Validation (Alz+Tempal)	787	83	49	58.92%	41.08%

Table 19. Comparing accuracy of prediction techniques using TempAl on Synthetic datasets.

Dataset(Learning Algorithm)	Train	Test	Correct	Prediction Accuracy	Prediction Error
Synthetic (Alz)	100	1	1	100%	0%
Synthetic (Alz+Tempal)	100	1	1	100%	0%
Synthetic (Alz)	100	10	10	100%	0%
Synthetic (Alz+Tempal)	100	10	10	100%	0%
Synthetic (Alz)	1400	90	89	98.88%	1.12%
Synthetic (Alz+Tempal)	1400	90	90	100%	0%
Synthetic (Alz)	13905	1544	1532	99.22%	0.78%
Synthetic (Alz+Tempal)	13905	1544	1532	99.22%	0.78%
Cross Validation (Alz)	13905	1544	1292	83.68%	16.32%
Cross Validation (Alz+Tempal)	13905	1544	1292	83.64%	16.36%

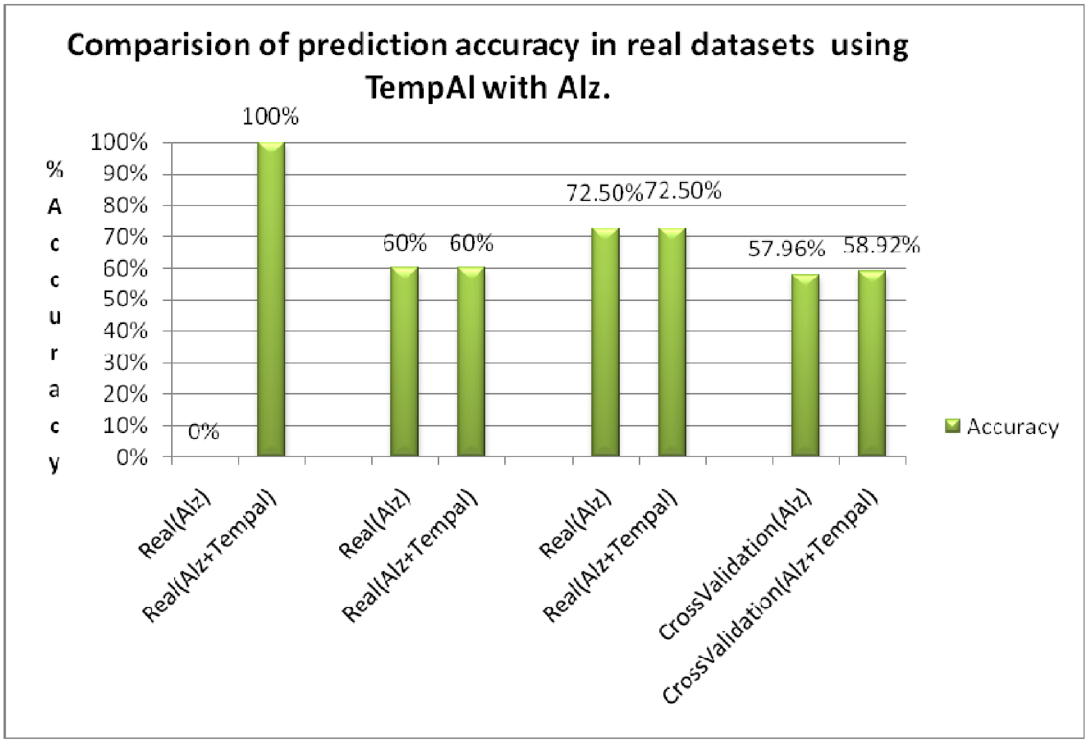


Figure 33. Percentage accuracy in real datasets in prediction experiment using Alz with TempAl.

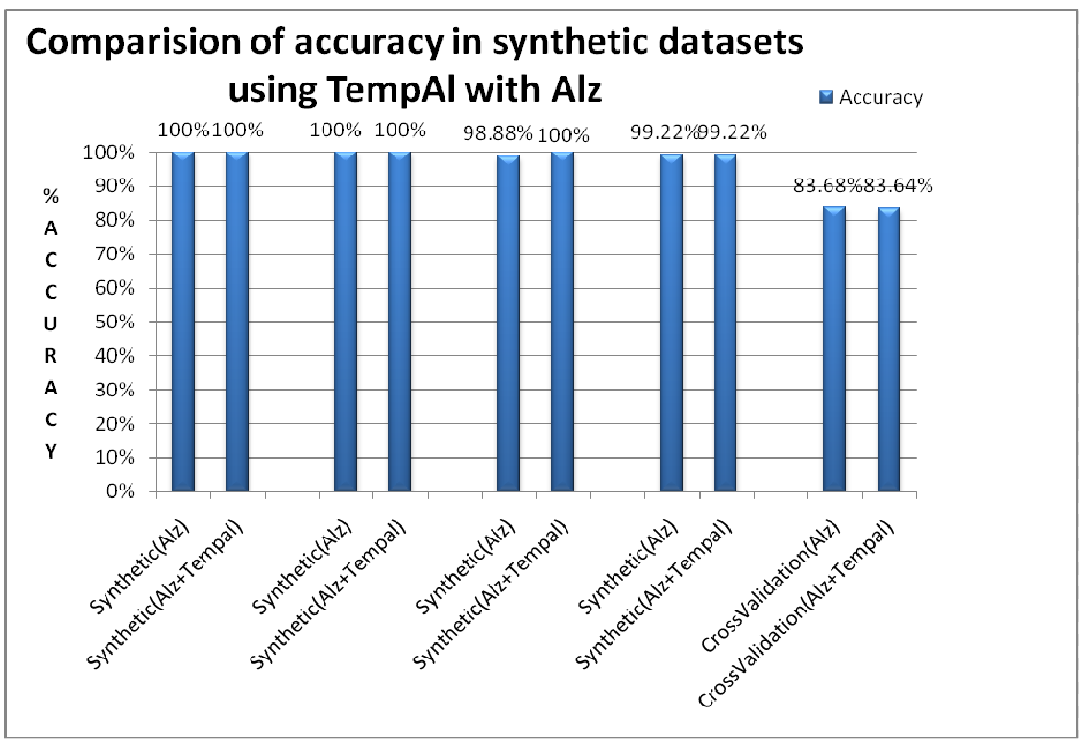


Figure 34. Percentage accuracy in synthetic datasets in prediction experiment using Alz with TempAl.

We see that Tables 18 and 19 present us with results observed in the prediction experiment. We need to note that accuracy values are computed as the ratio of the count of number of correct predictions to the total number of predictions. The cross validation performed on these datasets is k-fold cross validation [65]. For this we randomly divide the original dataset into 10 partitions and use 9 partitions for training and the last partition for testing. This process is repeated for 10 folds and the average of the error noted is reported as the error.

We observe that the Alz enhanced with TempAl did perform similarly to the original Alz-TDAG technique. This particular dataset did not make particular use of temporal relationships. To illustrate the type of situation in which temporal analysis will specifically aid event prediction we test TempAl on a carefully constructed test case which is described next.

Test Case Scenario:

We observe that the previous datasets do not highlight the true potential of leveraging temporal relations for enhancing prediction. Thus we developed a scripted test case to observe as how the temporal relations would help make a better prediction. Let us look at a small example where temporal information does enhance prediction. Let us consider the example where the following events occur in the given sequence shown as follows : (a ON), (a OFF), (a ON), (b ON), (a ON), (b ON), (b ON), (b ON), (b ON), (b ON),(a ON), (a ON), (b ON), (c ON), (c ON), (d ON), (d ON),(c ON),(b ON),(a ON),(c OFF), (a, OFF). In this scenario the next event that will occur is (a ON). Now we see that when we run this training set on Alz and then load the test set and we see that it predicts "b" to be the next. We see that this is an incorrect prediction. Now let us run the same experiment using Alz with TempAl and we see that on the test set it correctly predicts a as the next event. Although (b ON) occurs most often overall, the temporal relationship of (a ON) AFTER (a OFF) is prevalent and should ultimately influence the predictor to output (a ON) as the most likely next event to occur. Thus when we leverage the temporal relations we can enhance the prediction and therefore it would aid to improve the prediction accuracy. Let us look at this scenario in more detail. Table 20 gives us a description of the training set, test set and temporal relations formulation set.

Table 30. Training, Test Set, Temporal relations for test case scenario

Training Set: a ON a OFF a ON b ON a ON b ON b ON b ON b ON b ON a ON a ON b ON c ON c ON d ON d OFF c ON b ON a ON c OFF a OFF	a BEFORE b, a BEFORE a, a BEFORE b, a BEFORE c, a BEFORE d, a BEFORE c, a BEFORE a, a AFTER a, a OVERLAPS b, a BEFORE b, a BEFORE b, a BEFORE a, a BEFORE b, a BEFORE c, a BEFORE d, a BEFORE c, a BEFORE a, b AFTER a, b OVERLAPPEDBY a, b MEETS b, b BEFORE b, b BEFORE a, b BEFORE b, b BEFORE c, b BEFORE d, b BEFORE c, b BEFORE a, b AFTER a, b AFTER a, b AFTER b, b METBY b, b BEFORE a, b BEFORE b, b BEFORE c, b BEFORE d, b BEFORE c, b BEFORE a, a AFTER a, a AFTER a, a AFTER b, a AFTER b, a AFTER b, a BEFORE b, a BEFORE c, a BEFORE d, a BEFORE c, a BEFORE a, b AFTER a, b AFTER a, b AFTER b, b AFTER b, b AFTER b, b AFTER a, b CONTAINS c, b CONTAINS d, b OVERLAPS c, b BEFORE a, c AFTER a, c AFTER a, c AFTER b, c AFTER b, c AFTER b, c AFTER a, c DURING b, c BEFORE d, c BEFORE c, c BEFORE a, d AFTER a, d AFTER a, d AFTER b, d AFTER b, d AFTER b, d AFTER a, d DURING b, d AFTER c, d BEFORE c, d BEFORE a, c AFTER a, c AFTER a, c AFTER b, c AFTER b, c AFTER b, c AFTER a, c DURING b, c AFTER c, c AFTER d, c FINISHES a, a AFTER a, a AFTER a, a AFTER b, a AFTER b, a AFTER b, a AFTER a, a AFTER b, a AFTER c, a AFTER d, a FINISHESBY c
Test Set: a ON d OFF	
Temporal Relations on Training Set: [NOTE: Temporal relations are formed on complete device cycle .i.e. complete cycle of a device from an ON to OFF or an OFF to ON state is pursued within a window of a single day to form an event for associating temporal relation with another event.] a BEFORE a, a BEFORE b, a BEFORE b,	

Given in Table 20, when we use Alz-TDAG we see that it calculates b as the most likely event based on overall frequency without temporal relationship information, resulting in an incorrect prediction. When we incorporate temporal relations into the probability calculation we see that it correctly predicts (a ON) as the next event. On the other hand, it later fails to predict event (d OFF) because it did not occur significantly anywhere in the training data, thus providing weaker temporal information. Thus this simple example stands as an illustration to check the performance of TempAl and Alz. Figure 35 shows a screenshot of the raw output collected from Alz on the test case and Figure 36 shows a screenshot of the raw output collected from TempAl + Alz prediction.

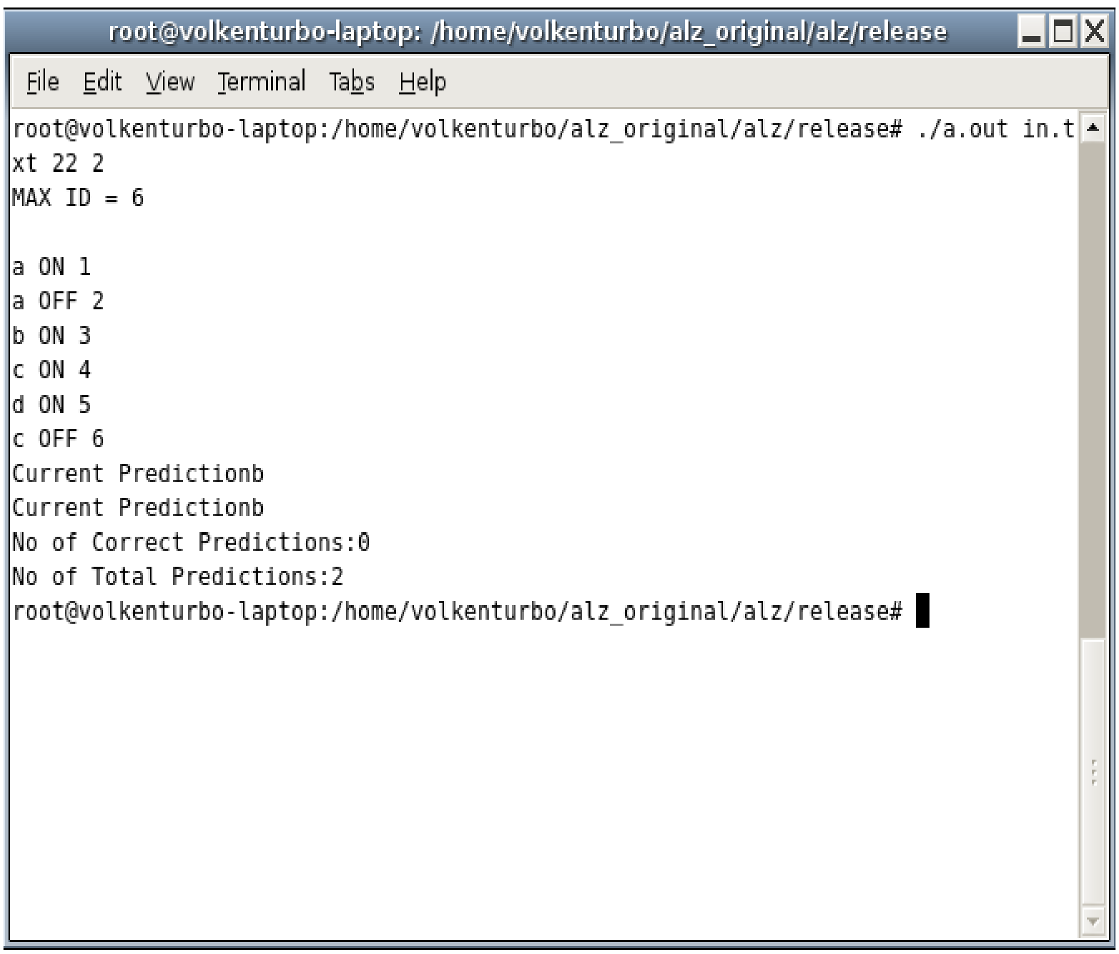


Figure 35. Raw output on the test case dataset using Alz

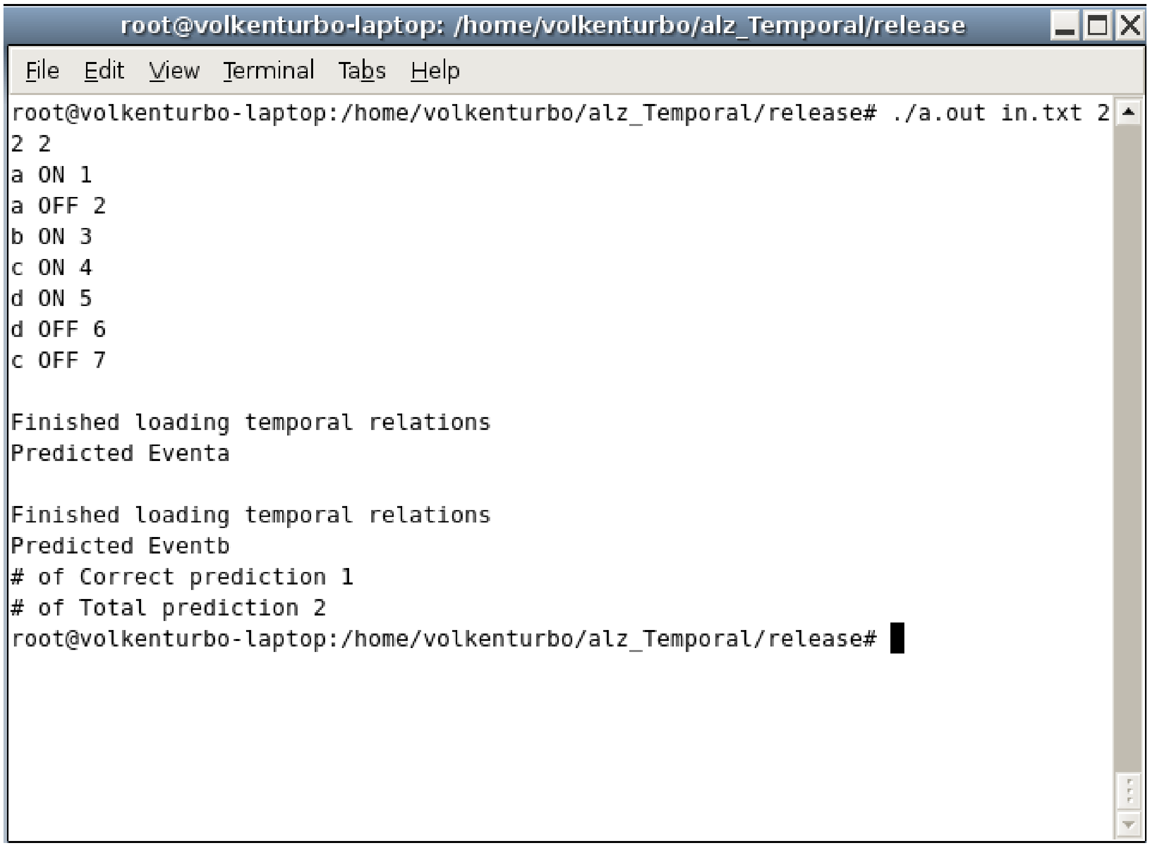


Figure 36. Raw output on the test case dataset using Alz + TempAl.

Discussion on Enhancing Prediction by Adding Temporal Relations based Probability to ALZ

This experiment [63] differs from the earlier experiment vastly. In the earlier prediction experiment we used rule based prediction, where we generated rules where the antecedent of a rule is used to predict the consequent of the rule. This current experiment uses the temporal information to calculate the probability of next event to occur and leverages the existing sequential prediction technique by adding temporal information. The dataset used for the experiment plays a major role with the prediction experiments. We note that the main reason for a significant error rate is the amount of data used, which is small and covers a smaller set of training examples. As we have larger datasets we see that the performance of the temporal relations enhanced prediction would also improve drastically over time. Tables 18 and 19 summarize the observed accuracy of the prediction performance on real and synthetic datasets.

Another important point to discuss is that Alz stores observed events with frequencies in a trie. The temporal relations can also be stored using a graph based approach where events are related by a temporal relation and the weight of the link or relation is the frequency of its occurrence. This approach can be further investigated as future work.

Activity visualization using Time Intervals

The visualizer tool visualizes the identified activity intervals over time and is applicable to dynamic scenarios with real time data streams. It gives the resident complete control of visualization with scrolling and enabled to look for patterns. A screenshot of this tool is displayed in Figure 37. In this illustration, we can visualize the various activities with their event IDs, the name of the activity/sensor, its start time and end time and the corresponding time interval. Finally, this tool can acts as an Integrated Interface for visualization for search and discovery of temporal patterns with smart home datasets. We apply the longest subsequence search technique to find any interesting pattern. The reason we apply this technique is that we are apply it for patterns identified over a window of size of a single day and they would be interesting only when we try to look for patterns in a single day and compare those patterns to the following day or any other day. An illustration of how the tool would look is given in Figure 37. Additional details of the visualizer are provided in Appendix B.

Figure 37. Interval Visualization Tool: A close-up screenshot which shows the daily events and how they are scaled and represented. Event times are also displayed in a scrollable display which allows for observable comparison between days as well as between events that occurred on the same day.

Some interesting discovered patterns are reported in Table 21. Table 22 gives the total number of interesting temporal patterns which were identified in the dataset.

Table 21. Interesting patterns found using the pattern recognition tool.

Date	Date	LCS Identified
3/2/2003	3/3/2003	g11equalg12
3/3/2003	3/4/2003	d9equald9
3/4/2003	3/5/2003	No
3/5/2003	3/6/2003	No
3/6/2003	3/7/2003	d9equald9
3/7/2003	3/8/2003	d9equald9

Table 22. Number of patterns found in real experiment dataset.

	No # Patterns Found	No # Days
Real Data	18	66

In Table 17 we see a few interesting patterns which occur across multiple days. We note that the g11 is a lamp and g12 is a lamp. We also see that they are turned on at the same time. We report a sample of such identified relations in Table 17. One of the reasons to have d9 equals d9 is due to the fact that the events occur in millisecond time intervals and the sensor would either round off or record the time to the nearest second and thus would lead to have multiple events occurring with the same timestamp. Table 18 reports that there were 18 different reoccurring patterns found in total 66 days. The next step to this representation would be enabling the resident to directly visualize the interesting temporal relations based patterns using a bubble plot view.

Summary

In this chapter we present experimental results which evaluate anomaly detection and prediction algorithms which are enhanced using temporal relation analysis. We also present visualization and pattern recognition evaluates results. In the next chapter we summarize our conclusions and discuss future directions for this research.

CHAPTER SIX
CONCLUSIONS AND DIRECTIONS FOR FUTURE WORK

Research has shown us that older Americans prefer an independent lifestyle [66]. With a growing aging population that desires to maintain their independent living styles and the increase in healthcare costs, the need for, smart and cost effective homes arises.

Conclusion

Smart environments are essential today, because of the feasible technology and networked computing, and also the need for home based healthcare and assistance rapidly rising. In this work, we have proposed a technique for the discovery of temporal rules in event sequences in a smart home. The aim of this study was to show the feasibility of leveraging temporal relations in activities in a smart environment and to propose a methodology for prediction and anomaly detection. The approach suggests that in cases where the event information is too general, it is possible to expose it using temporal interval representation and applying temporal relations. We have described an approach using temporal relations to detect anomalies, aid prediction and also look for interesting patterns. We have shown that temporal relations between events can be used effectively for smart home and smart environment domain problems. Now with anomaly detection, some anomalies may be detected without significant use of resources or techniques and for some additional techniques be needed based on resident (say the resident is an elderly individual and may have a very fixed pattern of events or if a teenager, which results in irregular activity pattern).

The presented approach is a novel approach from a theoretical point of view and also the preliminary results seem promising. Obviously, parts of the method need some more polishing, and the need to extend the study to a larger data set for very promising results is clearly visible. Provided we can collect more data, it would be easy to improve the model by (at least local) optimization on the space of possible rules. We hope that the measures of temporal information we have used will help in all aspects, but we are also planning to further investigate the temporal relations properties and that of other candidate measures not considered here for this current study. In this work we presented an approach to temporal pattern mining. One application of such is the prediction of events by using (temporal) association rules and incorporating temporal information. Besides evaluation future work on larger datasets one has to address further ways to reduce complexity of these techniques.

Temporal reasoning enhances data mining in smart environments by adding information about expected temporal interactions between resident activities. Based on our study, we conclude that the use of temporal relations provides us with a new approach for anomaly detection. We tested our algorithm on relatively small datasets, but will next target larger datasets with real activity data collected. Other future directions of this work also include improving activity prediction using temporal relations in smart home data. One challenge this work introduces is determining which observed events belong to the same activity (say we have two lamp events back to back, the problem of grouping them as one or should be include them to be separate), and thus the same temporal interval. In this study we grouped events that turned a device on together with those that turned the same device off. However, for a more extensive study we need to determine a general method for grouping events.

Temporal rule based pattern analysis is a niche area in temporal mining world. We notice that the use of temporal relations provides us with a unique approach for anomaly detection. We will also expand the temporal relations by including more temporal relations, such as until, since, next, and so forth, to create a richer collection of useful temporal relations.

The goal of the association rule mining based approach for prediction is to generate a rule-based prediction system, which can be integrated into a comprehensive smart home architecture. We use the most recent observed event to identify which rules to use for prediction. Once the rule or a set of rules is identified then the rules are used for prediction. This approach showed some encouragement to use association rule mining to enhance prediction. Some disadvantages of this system include the identifying interesting rules and also handling multiple rules with safe confidence levels. Some possible solutions to the above mentioned problem are discussed in the future work section.

The next prediction experiment involved a method of enhancing an existing sequential prediction technique by incorporating temporal information to improve prediction performance. We see that the fusion of the information is intuitively appropriate as the sequential prediction uses a trie-based prediction algorithm and this implicitly incorporates the temporal relation "before" and uses order-based analysis for computing the prediction probability. Now we also incorporate temporal information into event probability calculates at context sizes greater than 0 because at the higher orders in the phrase we have all the temporal information which would make it richer than the single existing "before" relation. Evaluation of this combined prediction approach shows encouraging results and opens the field to new ideas such as considering graph-based approaches and link analysis approaches for prediction in smart environment domains.

Finally, it is worth remembering that human activities are need based and are thus clouded with the resident's emotional state and the physical energy required for events to be performed. As a result, smart home adaptive automation is by itself a difficult task, with potentially a lot of disagreement between multiple residents or the influence of a single resident through process. For now, our work have is bound by a single resident. We therefore have no measure of inter-inhabitant or multi-resident agreement which could serve as an upper bound of the performance of this system, although we are currently planning and setting up this smart environment to do this at a larger scale.

Future work

A major enhancement to this work could consist of inclusion of interval analysis for intermediate states of devices. Intermediate states are those which exist between an ON and OFF state, for instance, we have a lamp controlled by a dimmer switch, we can see that there can be a maximum of dimmer adjustments or levels that can be supported by the dimmer, which form the intermediate states. Currently the interval formation only handles ON and OFF states of the devices and would include other intermediate states in the future work.

Another major enhancement in the future would be investigating techniques to identify the cut-off or thresholds which could replace the existing cut-off evaluator which is the sum of average and two times standard deviation of the probability data. There also exists some future work to find better fusion techniques to help use temporal based prediction and normal sequential prediction for optimal prediction, where optimal prediction is the prediction of greatest accuracy.

While making sense of sensor data can be challenging for smart environment applications, the problem is made even more complex when the environment houses more than one resident. To aid the capabilities of our temporal data mining, 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 work with for determining features of the data during the mining stages. In this case, we are enriching the data with information about entities moving within the space. 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 later associate these patterns to behavior models and use them for entity identification and resident profiling would be one direction for this work.

Agents in dynamic environments have to deal with changes over time. We establish temporal statement from the observation or prevision of changes. The relationship between space and time is the consequence of the observation of changes as the perception of spatial alternations denotes the existence of time. Temporal relations do not replace spatial relations but they just provide a different perspective and an integrated view supports the identification of both space and time. Enhancing this model for temporal to spatiotemporal models would be the next immediate step. As reasoning of space and time plays a vital role in our everyday lives and this can be scheduling our work or the events that occur in homes. And especially smart homes can support us with these reasoning capabilities for certain events and actions associated to certain space in the smart home. This would directly aid the elderly with cognitive impairments.

We did see an interesting application of association mining in the context of temporal relations in smart homes for prediction. For prediction we use the antecedent of a rule to predict the consequent of the rule. But for future work we should look at the fact that not all association rules may be suitable for prediction and the fusion of spatial knowledge with association rules would help decide interestingness in association rules and would help make better prediction. We can define the precision or interestingness of the rules found as follows: Precision = # of interesting rules discovered / # of total rules discovered. Methods to find trade-off between different set support and confidence should also be investigated. Thus the future works includes the development of appropriate metrics for rule quality and develop new techniques for rule post-processing. Regarding the rule mining itself, it is obvious that it is a computationally very costly process, and more work should be performed on how to optimize this part of the process. Another important future work would involve a moving window which would help track patterns on a shorter time and make the existing system more adaptable to the resident and changing patterns.

The prediction model which enhances ActiveLeZi based approach by incorporating temporal information needs more in-depth analysis by investigating fusion of graph based approaches with trie-based approach. Overall the temporal relations based prediction should investigate graph based and link prediction techniques.

The problem of anomaly detection can take different approaches. Currently we try to compute the evidence whether the current predicted event or the most recently occurred event is an anomaly or not. Future work could include adapting the temporal relations which aid anomaly detection based on resident history, for instance choosing among the nine anomaly detection aiding temporal relations the best one which could be used for anomaly detection. External events also contribute to anomalies in smart environment: one simple example would be a phone call which could trigger the resident into a thinking state and might turn on the lamp without his consent or conscious request. Emotional analysis techniques should also be investigated. Another future use of event prediction is its use for reminder assistance. If the resident queries the algorithm for the next routine activity, the activity or activities with the greatest probability will 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. We can in the future enhance the visualizer with embedding different pattern recognition techniques for identifying of more patterns of interest. We would also then evaluate as how effective is this tool to the resident when the patterns were identified and changed to enhance the resident's comforts or enrich the lifestyle.

Overall we need to note that the temporal relations based techniques need to be fully developed and should incorporate richer collection of temporal relations to avoid discarding many details but they still stand to encourage anomaly detection and prediction techniques and stand as an novel method in smart environment domain.

CHAPTER SEVEN

BIBLIOGRAPHY

1. James F. Allen, and George Ferguson, *Actions and Events in Interval Temporal Logic*, Technical Report 521, July 1994.

2. Guralnik V. and Srivastava J. *Event detection from time series data*. Proceedings of fifth ACM SIGKDD international conference on knowledge discovery and data mining, California, US, pp 33-42, 1999.

3. Herbert LE, Scherr PA, Bienias JL, Bennett DA, and Evans DA. *Alzheimer's disease in the US population: Prevalence estimates using the 2000 census*. Archives of Neurology 2000; 60:119-1122.

4. Vikramaditya Jakkula & Diane J. Cook. *Learning temporal relations in smart home data*. Proceedings of the second International Conference on Technology and Aging. Canada, 2007.

5. Cook, Diane; Das, Sajal (2004). *Smart Environments: Technology, Protocols and Applications*. Wiley-Interscience. ISBN 0-471-54448-5.

6. S.K. Das, and D. J. Cook, *Smart home environments: A paradigm based on learning and prediction*. In Wireless Mobile and Sensor Networks: Technology, Applications and Future Directions. Wiley, 2005.

7. D. J. Cook, M. Youngblood, E. Heierman, K. Gopalratnam, S. Rao, A. Litvin, F. Khawaja. *MavHome: An Agent-Based Smart Home*. Proceedings of the IEEE

International Conference on Pervasive Computing and Communications. 521-524 (2003).

8. G. Michael Youngblood. *Automating Inhabitant Interactions in Home and Workplace Environments through Data-Driven Generation of Hierarchical Partially-Observable Markov Decision Processes*. Doctoral Dissertation. The University of Texas at Arlington. August 2005.

9. M. Morris, S. S. Intille, and J. S. Beaudin. *Embedded Assessment: overcoming barriers to early detection with pervasive computing*. Proceedings of PERVASIVE 2005 Berlin Heidelberg: Springer-Verlag, 2005.

10. S. Consolvo, P. Roessler, B.E. Shelton, A. LaMarca, B. Schilit, & S. Bly. IEEE Pervasive Computing Mobile and Ubiquitous Systems: *Successful Aging*, Vol. 3, No. 2, (Apr-Jun 2004), pp. 22-29.

11. G. Michael Youngblood, Lawrence B. Holder, and Diane J. Cook. *Managing Adaptive Versatile Environments*, Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom).(2005)

12. K. Gopalratnam and D. J. Cook. *Active LeZi: An Incremental Parsing Algorithm for Sequential Prediction*. International Journal of Artificial Intelligence Tools, 14(1-2):917-930, 2004.

13. Vikramaditya Jakkula and Diane J. Cook. *Learning temporal relations in smart home data*. Proceedings of the second International Conference on Technology and Aging, Canada, June 2007.

14. G. Jain, D. Cook, and Vikramaditya Jakkula. *Monitoring health by detecting drifts and outliers for a smart environment resident*. Proceedings of the International Conference On Smart Homes and Health Telematics, 2006.

15. D. J. Cook, S. Das, Karthik Gopalratnam, and Abhishek Roy. *Health Monitoring in an Agent-Based Smart Home*. Proceedings of the International Conference on Aging, Disability and Independence Advancing Technology and Services to Promote Quality of Life, 2003.

16. S. Das and D. J. Cook. *Health Monitoring in an Agent-Based Smart Home*. Proceedings of the International Conference on Smart Homes and Health Telematics *(ICOST)*, Singapore, September, 2004.

17. Vikramaditya Jakkula, and Diane J. Cook. *Prediction Models for a Smart Home based Healthcare System.* Proceedings of the 21st IEEE international conference on Advanced Information Networking and Applications, May 2007.

18. Vikramaditya Jakkula, Michael G. Youngblood and Diane J. Cook. *Identification of Lifestyle Behaviors patterns with Prediction of the Happiness of an Inhabitant in a Smart Home.* AAAI Workshop on AAAI Workshop on Computational Aesthetics: Artificial Intelligence Approaches to Beauty and Happiness, Boston, July 2006.

19. S. Intille, J. Herigon, W. Haskell, A. King, J. A. Wright, and R. F. Friedman. *Intensity levels of occupational activities related to hotel housekeeping in a sample of minority women*. Proceedings of the Annual Meeting of the International Society of Behavioral Nutrition and Physical Activity, 2006.

20. J.S. Beaudin, S.S. Intille, and M. Morris. *Micro Learning on a Mobile Device.* Proceedings of UbiComp 2006 Extended Abstracts (Demo Program), 2006.

21. J. Nawyn, S. S. Intille, and K. Larson. *Embedding behavior modification strategies into a consumer electronics device: a case study*. Proceedings of UbiComp 2006 , Springer-Verlag, 2006.

22. J. Ho and S. S. Intille. *Using context-aware computing to reduce the perceived burden of interruptions from mobile devices*. Proceedings of CHI 2005 Connect: Conference on Human Factors in Computing Systems. New York, NY: ACM Press, 2004.

23. Gary Look and Howard Shrobe. *Towards Intelligent Mapping Applications: A Study of Elements Found in Cognitive Maps*. In IUI '07: Proceedings of the 12th International Conference on Intelligent User Interfaces, pp.309--312. New York, NY, USA, 2007.

24. S. Consolvo, P. Roessler, & B.E. Shelton. *The CareNet Display: Lessons Learned from an In Home Evaluation of an Ambient Display*. Proceedings of the 6th Int'l Conference on Ubiquitous Computing: UbiComp '04, (Sep 2004), pp. 1-17.

25. S. Consolvo, P. Roessler, B.E. Shelton, A. LaMarca, B. Schilit, & S. Bly. *Technology for Care Networks of Elders*. IEEE Pervasive Computing Mobile and Ubiquitous Systems: Successful Aging, Vol. 3, No. 2, (Apr-Jun 2004), pp. 22-29.

26. S. Consolvo, J. Towle. *Evaluating an Ambient Display for the Home*. In Extended Abstracts of Human Factors in Computing Systems: *CHI '05*, (Apr 2005).

27. Majd Alwan, Steve Kell, Siddharth Dalal, Beverely Turner, David Mack and Robin Felder. *In-Home Monitoring System and Objective ADL Assessment: Validation Study.* International Conference on Independence, Aging and Disability, 2003.

28. Tracy Barger, Donald Brown, Majd Alwan. *Health Status Monitoring Through Analysis of Behavioral Patterns.* Lecture Notes in Artificial Intelligence (LNCS/LNAI), Proceedings of the 8th congress of the Italian Association for Artificial Intelligence (AI*IA) on Ambient Intelligence, Springer-Verlag, Pisa, Italy, September 2003.

29. Majd Alwan, David Mack, Siddharth Dalal, Steve Kell, Beverely Turner, Robin Felder. *Impact of Passive In-Home Health Status Monitoring Technology in Home Health: OutcomePilot.* Proceedings of the Tran disciplinary Conference on Distributed Diagnosis and Home Healthcare (D2H2), 2 - 4 April 2006, Arlington, VA.

30. A. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen. *The Gatortech smart house: A programmable pervasive space.* IEEE computer, 38(3):50-60, 2005.

31. William C. Mann, Sumi Helal. *Pervasive Computing Research on Aging, Disability and Independence,* 2005.

32. M. E. Pollack, C. E .McCarthy, S. Ramakrishnan, I. Tsamardinos, L. Brown, S. Carrion, D. Colbry, C. Orosz, and B. Peintner. Autominder: *A Planning, Monitoring, and Reminding Assistive Agent*. 7th International Conference on Intelligent Autonomous Systems, March, 2002.

33. M. E. Pollack, L. Brown, D. Colbry, C. E. McCarthy, C. Orosz, B. Peintner, S. Ramakrishnan, and I. Tsamardinos. Autominder: *An Intelligent Cognitive Orthotic System for People with Memory Impairment.* Robotics and Autonomous Systems, 44:273-282, 2003.

34. M. E. Pollack. *Opportunities and Challenges in Assistive Technology for Elders*. Testimony presented to the U.S. Senate Committee on Aging, Apr. 27, 2004.

35. F. Doctor, H. Hagras, and V. Callaghan. *A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabitant environment*. IEEE Transactions on Systems, Man, and Cybernetics, Part A, 35(1): 55-65, 2005.

36. Tamara L. Hayes, Misha Pavel, Pamela K. Schallau, and Adriana M. Adami. *Unobtrusive Monitoring of Health Status in an Aging Population*. UbiHealth 2003: The 2nd International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications, 2003.

37. Mörchen, F. *A better tool than Allen's relations for expressing temporal knowledge in interval data.* In Proceedings the Twelveth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, (2006).

38. Björn Gottfried, Hans W. Guesgen, and Sebastian Hübner. *Spatiotemporal Reasoning for Smart Homes.* Designing Smart Homes, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Pg 16-34, Volume 4008/2006, July 2006.

39. V. Ryabov, Seppo Puuronen. *Probabilistic Reasoning about Uncertain Relations between Temporal Points*. Eighth International Symposium on Temporal Representation and Reasoning (TIME'01), 2001.

40. Hornsby, K.H. and Worboys, M.F. *Event-oriented approaches in geographic information science*. A special issue of Spatial Cognition and Computation 4(1), Lawrence Erlbaum, Mahwah, NJ, ISBN: 0-8058-9531-0, 2004.

41. Alex Dekhtyar, Robert Ross AND V.S. Subramanian. *Probabilistic temporal databases, I: algebra*, ACM Transactions on Database Systems (TODS), Volume 26, Issue 1 (March 2001).

42. Allen, J.F. and Ferguson, G. *Actions and Events in Interval Temporal Logic*. J. Logic and Computation 4, 5, 1994.

43. Allen, J.F. *Time and time again: The many ways to represent time*. Int'l. Jr. of Intelligent Systems 6, 4, 341-356, July 1991.

44. M. Youngblood, L. Holder, and D. Cook. *Learning Architecture for Automating the Intelligent Environment*. Proceedings of the Conference on Innovative Applications of Artificial Intelligence, 2005.

45. E. Heierman, M. Youngblood, and D. Cook. *Mining Temporal Sequences to Discover Interesting Patterns*. KDD Workshop on Mining Temporal and Sequential Data, 2004.

46. K. Gopalratnam and D. Cook. *Active LeZi: An Incremental Parsing Algorithm for Device Usage Prediction in the Smart Home.* Proceedings of the Florida Artificial Intelligence Research Symposium, 2003.

47. E. Heierman and D. Cook, Improving Home Automation by Discovering Regularly Occurring Device Usage Patterns, Proceedings of the International Conference on Data Mining, 2003.

48. K. Gopalratnam and D. Cook, Online Sequential Prediction Via Incremental Parsing: The Active LeZi Algorithm, IEEE Intelligent Systems, 22(1), 2007.

49. D. Cook and S. K. Das, How Smart are our Environments? An Updated Look at the State of the Art, Journal of Pervasive and Mobile Computing, 2007.

50. Allen, J.F. and P.J. Hayes. ``Moments and Points in an Interval-Based Temporal Logic." Computational Intelligence, January 1990.

51. Vikramaditya Jakkula and D. Cook, Temporal pattern discovery for anomaly detection in smart homes, Proceedings of the International Conference on Intelligent Environments, 2007.

52. Rakesh Agrawal and Ramakrishnan Srikant, *Fast Algorithms for Mining Association Rules*, Proc. 20th Int. Conf. VeryLarge Data Bases, Morgan Kaufmann,487--499, 1994.

53. Agrawal, R. and Srikant, R. (1995). Mining Sequential Patterns. In Proceedings of the 11th International Conference on Data Engineering, pages 3-14.

54. Ian H. Witten, Eibe Frank. Data Mining: Practical Machine Learning Tools and Techniques, 2nd Edition. Morgan Kaufmann, San Francisco. (2005).

55. A. Bhattacharya and S.K. Das, LeZi-Update: An Information-theoretic framework for personal mobility tracking in PCS networks, in *ACM/Kluwer Wireless Networks Journal*, vol. 8, no. 2-3 (2002), 121-135.

56. Wikipedia, et al. Longest Common Subsequence Problem. *Wikipedia*. Wikipedia Foundation, Inc. Retrieved May 25, 2007. http://en.wikipedia.org/wiki/Longest-common_subsequence_problem.

57. Wikibooks, et al. Algorithm Implementation Strings Longest Common Subsequence. *Wikibooks*. Wikimedia Foundation, Inc. Retrieved May 25, 2007. http://en.wikibooks.org/wiki/Algorithm_implementation/Strings/Longest_common_subsequence

58. Netronic. VARCHART XGantt. *NETRONIC Software GmbH*. Retrieved April 2007. http://www.netronic.com/english/index_prod.html?set_x_gantt.html

59. Microsoft Corporation. Visual Studio. Microsoft Publishing. Retrieved May 28, 2007. http://msdn.microsoft.com/vstudio/

60. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to Algorithms, The MIT Press; 2nd edition, 2001.

61. Hand D, Mannila H, and Snyth P. Principles of Data Mining: A Bradford Book, The MIT Press, Cambridge, Massachusetts; London, England, 2001.

62. Vikramaditya Jakkula and D. Cook, Using temporal relations in smart home data for activity prediction, Proceedings of the ICML Workshop on the Induction of Process Models, 2007.

63. Vikramaditya Jakkula and D. Cook, Mining sensor data in smart environments for temporal activity prediction, Proceedings of the ACM KDD First International Workshop on Knowledge Discovery from Sensor Data, 2007.

64. T. K. Hareven. Historical Perspectives on Aging and Family Relations. 2001. Handbook of Aging and the Social Sciences. 5th Edition. 141-159.

65. Wikipedia, et al. Cross Validation. *Wikipedia*. Wikipedia Foundation, Inc. Retrieved May 25, 2007. http://en.wikipedia.org/wiki/Cross_validation.

66. Vikramaditya Jakkula, Diane J. Cook, and Aaron S. Crandall, Knowledge Discovery in Entity Based Smart Environment Resident Data Using Temporal Relations Based Data Mining, ICDM Workshop on Spatial and Spatio-Temporal Data Mining, Omaha, Nebraska, 2007.

APPENDIX A

THE MAVLAB

MavLab Description

Figure A.1 illustrates the MavLab which was used for real data collection for this work. This lab consisted of X10 and ArgusD networks deployed covering ten zones which are illustrated in Figures A.2 and A.3.

Figure A.1 MavLab in 250 Nedderman Hall at UTA [8].

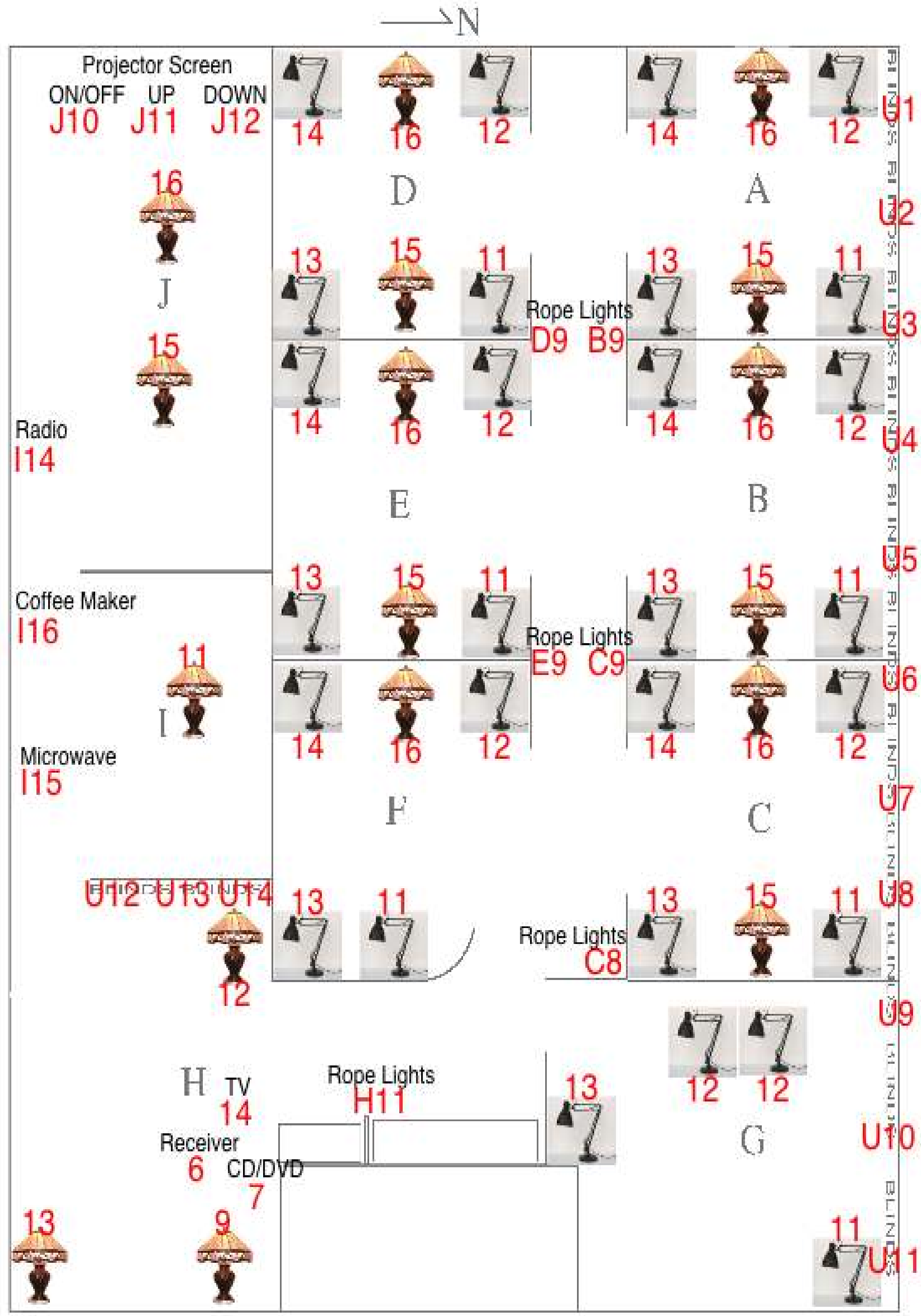


Figure A.2. MavLab configuration of X10 and ArgusMS actuators [8].

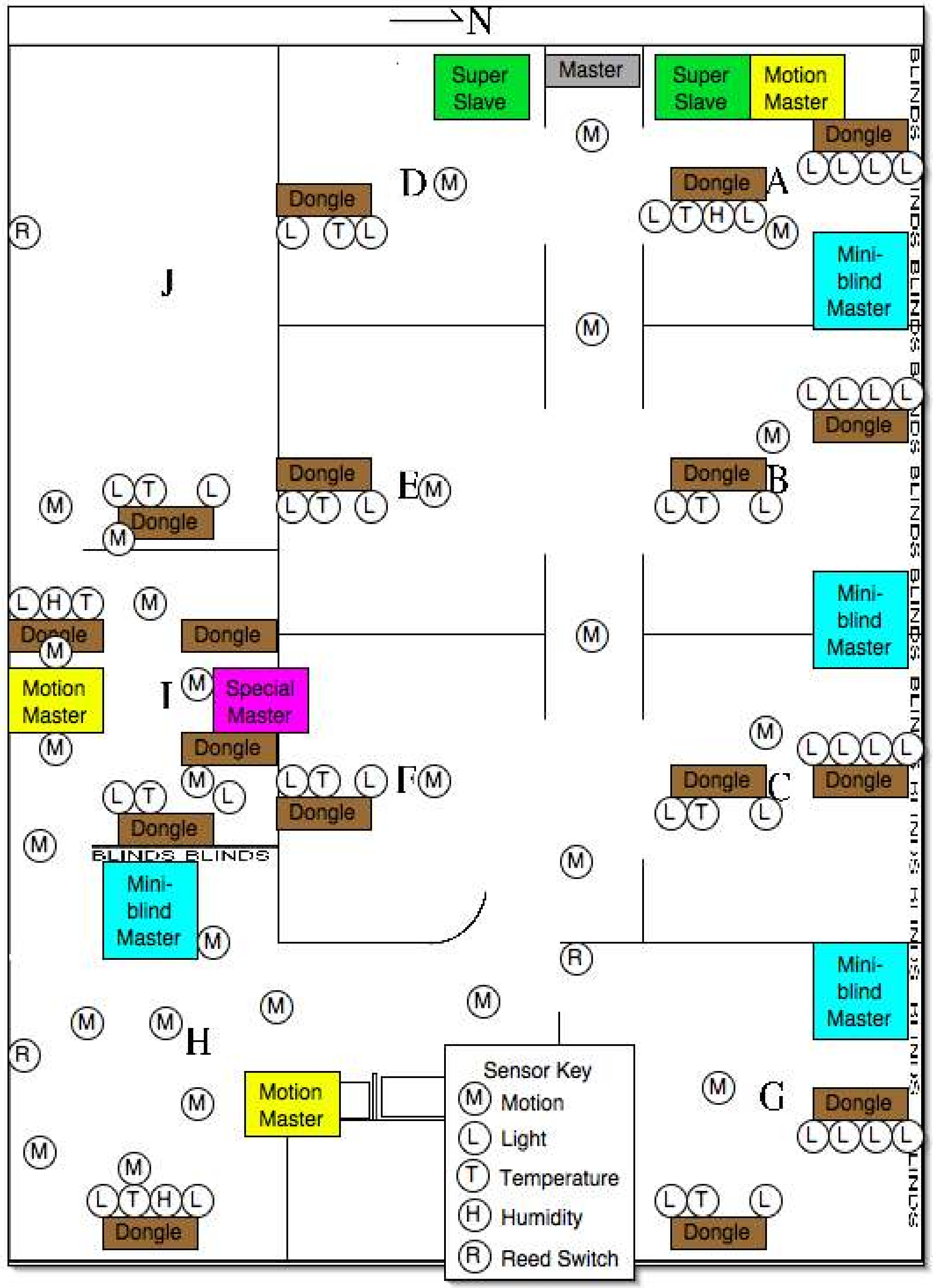


Figure A.3. MavLab sensor layout [8].

APPENDIX B

INPUTS, OUTPUTS & SCREENSHOTS

Data Generator

The data used for the experimentation consists of real and synthetic data. The real data collection environment is given in Appendix A and the synthetic data generation screenshot and sample of the generated output file is given here.

Figure B.1. Synthetic Data Generator Interface

Figure B.1 gives us an illustration of the data generator user interface. This was a simple interface which gives us the option to input the input file which contains the scenario description such as the number of devices, their various states, scenarios, and whether they occur either hourly, daily or weekly. A sample input file is given in Table B.2. Future work can focus on displaying this input file in the graphical user interface itself. This would help the data generator to generate the required synthetic data and would help us embed artificial scenarios for evaluation purposes. Figure B.3 illustrates the process of selecting an input file.

Table B.2. Synthetic Data Generator sample input file.

Sample Input File:
4
lamp 2 on off
thermostat 10 65 66 67 68 69 70 71 72 73 74
television 11 on off 2 3 4 5 6 7 8 9 10
oven 6 on off 15 30 45 60
1
E1 hourly 7 2 oven on NoOrder thermostat on Uniform 30
07/10/2002 09:30
off off 10 off off
10000

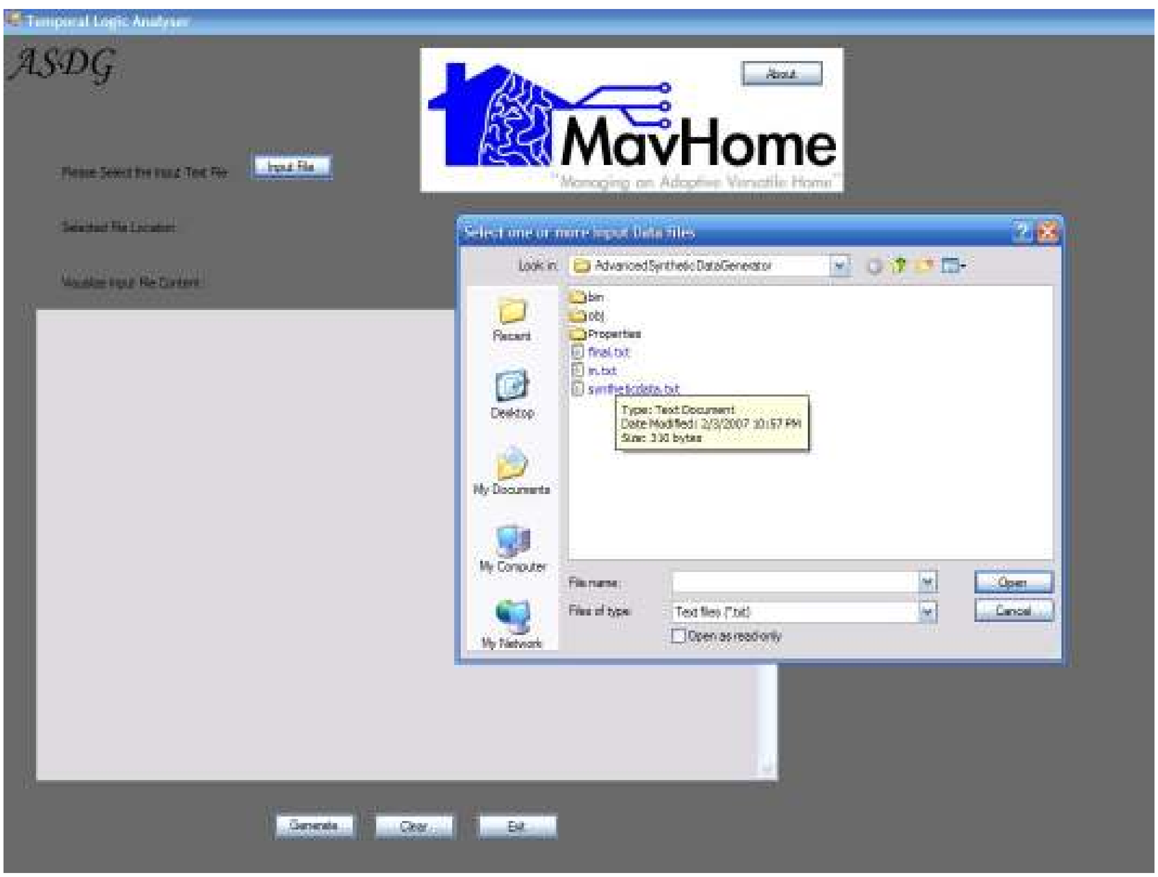


Figure B.3. Synthetic Data Generator Interface when the Input file button is clicked.

Figure B.4. Synthetic Data Generator Interface when the output is generated. The output is also written to a text file.

A sample of the generated data set is given in Table B.5.A. We see that randomness is incorporated into the time at which the devices are used and into the inhabitant's activities. We first created a synthetic data generator model user's pattern which consists of different activities comprising different locations (e.g., kitchen, bathroom) and involving interaction with a variety of devices (e.g., TV, lamp).

Additional details are provided in Chapter 3. The table B.5.B shows a sample real data collected from a smart environment.

Table B.5.A. Sample of the synthetic Data Generator output.

Timestamp	Device State	Device Name
2/1/2006 10:02	Off	Oven
2/1/2006 11:00	On	Lamp
2/1/2006 11:11	Off	Thermostat
2/1/2006 12:02	Off	Lamp
2/1/2006 12:35	Off	Cooker
2/1/2006 13:30	On	Lamp
2/1/2006 14:02	Off	Fan
2/1/2006 15:22	On	Oven
2/1/2006 16:09	Off	Oven
2/1/2006 16:13	Off	Lamp
2/1/2006 17:11	On	Lamp
2/1/2006 17:59	On	Fan
2/1/2006 18:23	Off	Lamp
2/1/2006 18:33	On	Cooker
2/1/2006 20:03	On	Thermostat
2/1/2006 20:14	Off	Cooker
2/1/2006 20:45	Off	Fan
2/1/2006 20:47	On	Oven
2/1/2006 21:12	On	Cooker
2/1/2006 21:20	Off	Cooker
2/1/2006 22:18	On	Fan
2/1/2006 23:03	Off	Thermostat
2/1/2006 23:14	On	Cooker
2/1/2006 23:45	Off	Fan
2/1/2006 23:47	Off	Oven
2/2/2006 00:12	Off	Cooker
2/2/2006 00:20	On	Cooker
2/2/2006 01:18	On	Fan
2/2/2006 02:03	On	Thermostat

Table B. 5.B. Sample of the real dataset collected from a smart environment

Timestamp	Location	Sensor Name	Sensor State
4/2/2003 10:36:0 PM	(Conference Room)	J11	ON
4/2/2003 10:51:0 PM	(Studio A)	A14	OFF
4/2/2003 11:2:0 PM	(Studio C)	C13	OFF
4/2/2003 11:6:0 PM	(Studio F)	F12	ON
4/3/2003 1:3:0 PM	(Living Room)	H9	ON
4/3/2003 1:4:0 PM	(Conference Room)	J10	ON
4/3/2003 1:8:0 PM	(Dining Room)	I14	ON

TempAl: Interval Analysis Tools

The experimentation tool consists of tools for analysis which are illustrated by these screenshots. The raw sensor data is processed for time intervals and later these intervals are used to find temporal relations between the data and used for knowledge discovery for anomaly detection and prediction.

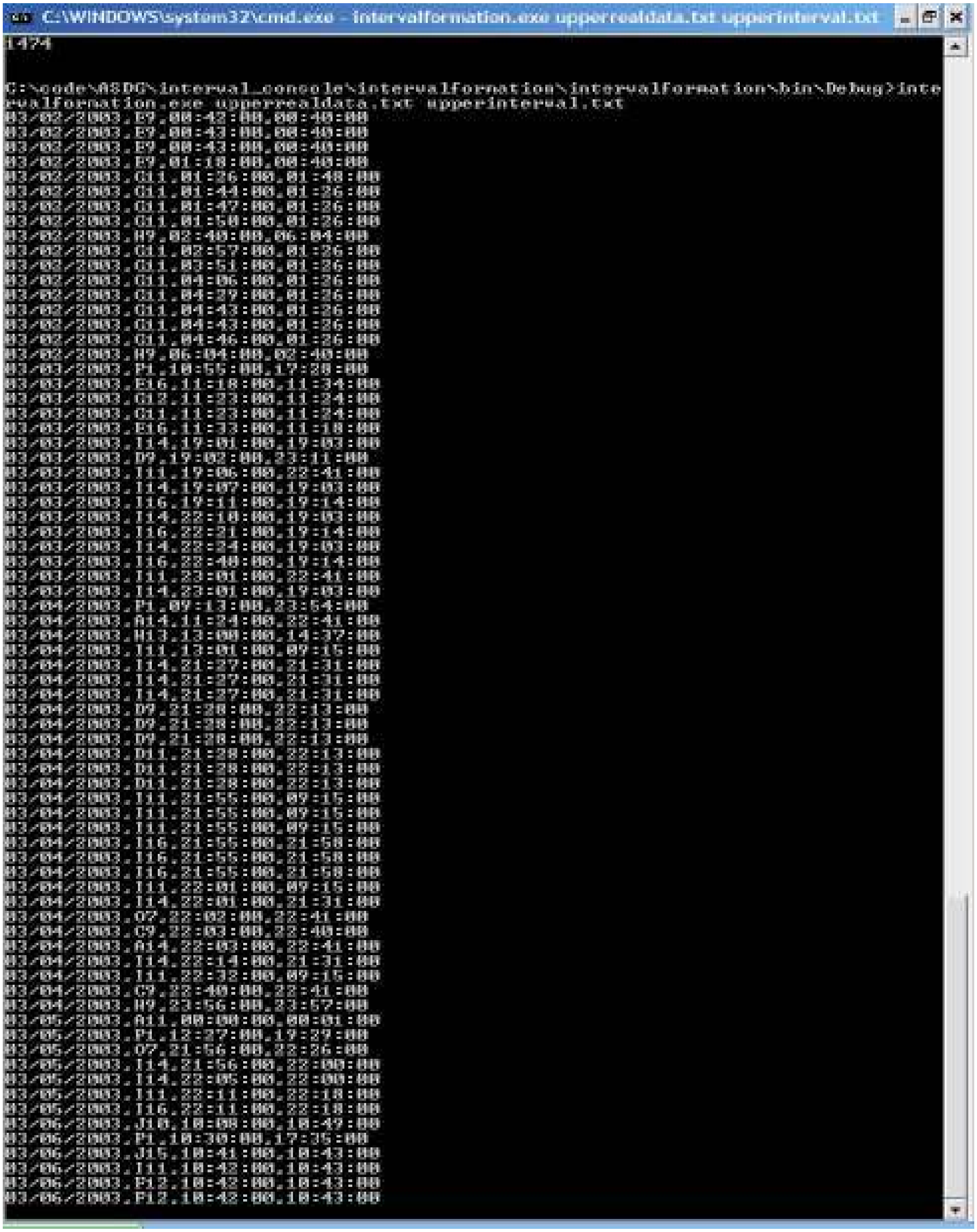


Figure B.6. Time Interval Formation screenshot.

Figure B.7. Temporal Relation Formulation screenshot.

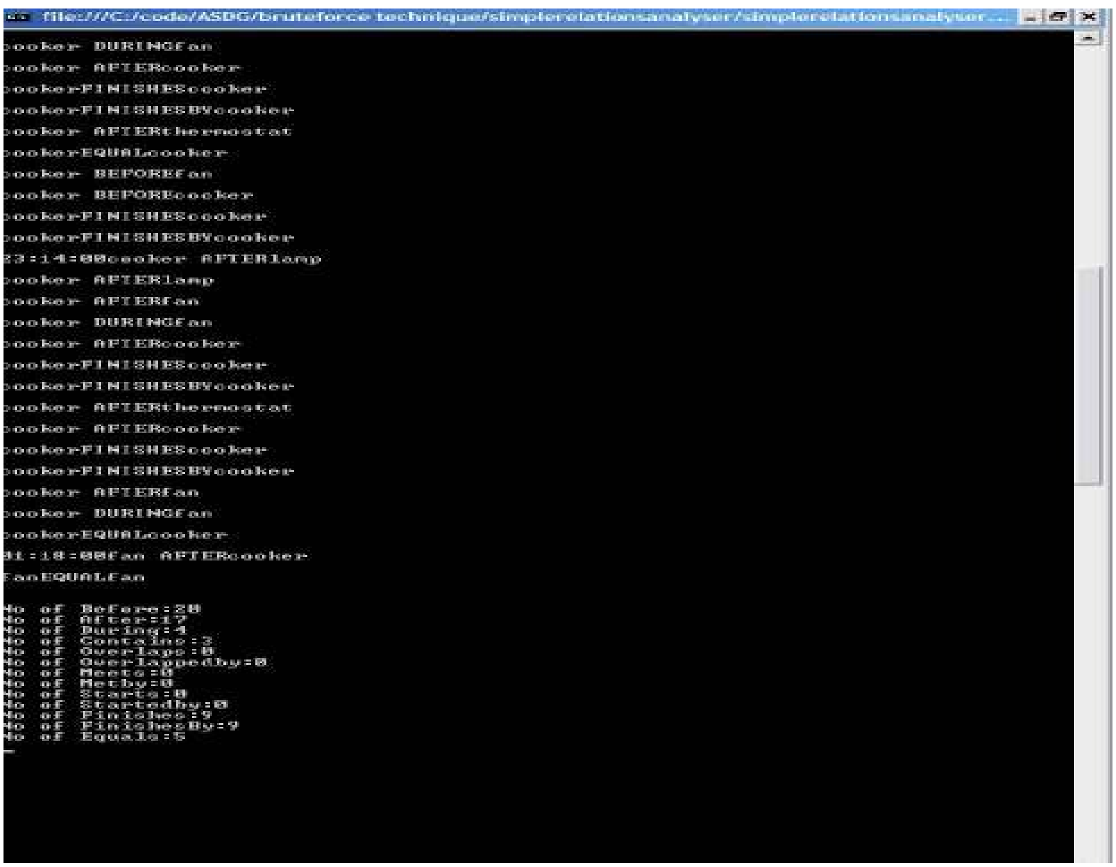


Figure B.8. Temporal Relation Formulation screenshot (Note: at the end a brief summary of the discovered rules is displayed).

TempAl: Anomaly Detection

The anomaly detection process looks whether the current event is an anomaly or not based on the temporal relations. Figure B.9 is a screenshot of the tool. This tool would provide an enhancement to the decision making system (ProPHeT) and would aid in making better decisions.

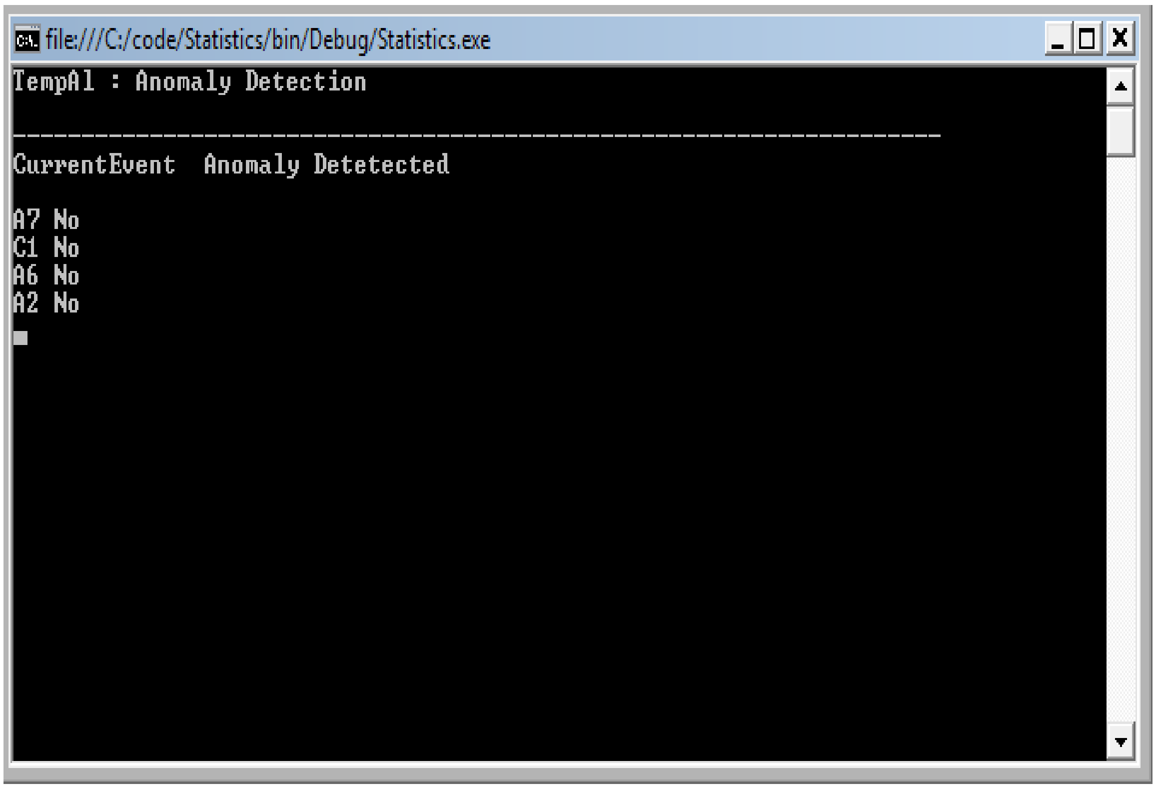


Figure B.9. Anomaly detection tool.

In Figure B.9 we see that the anomaly detection tool is illustrated and this enables us to know whether the current event is an anomaly or not. We note that it gives us the event name followed by whether it is computed to be an anomaly or not.

TempAl: Prediction

The prediction screenshots consist of the original sequential prediction output as well as the rule based and temporal relations enhanced prediction outputs. These are illustrated in Figures B.10, B.11, and B.12, respectively. We note that only the numbers of correct predictions are outputted at the end.

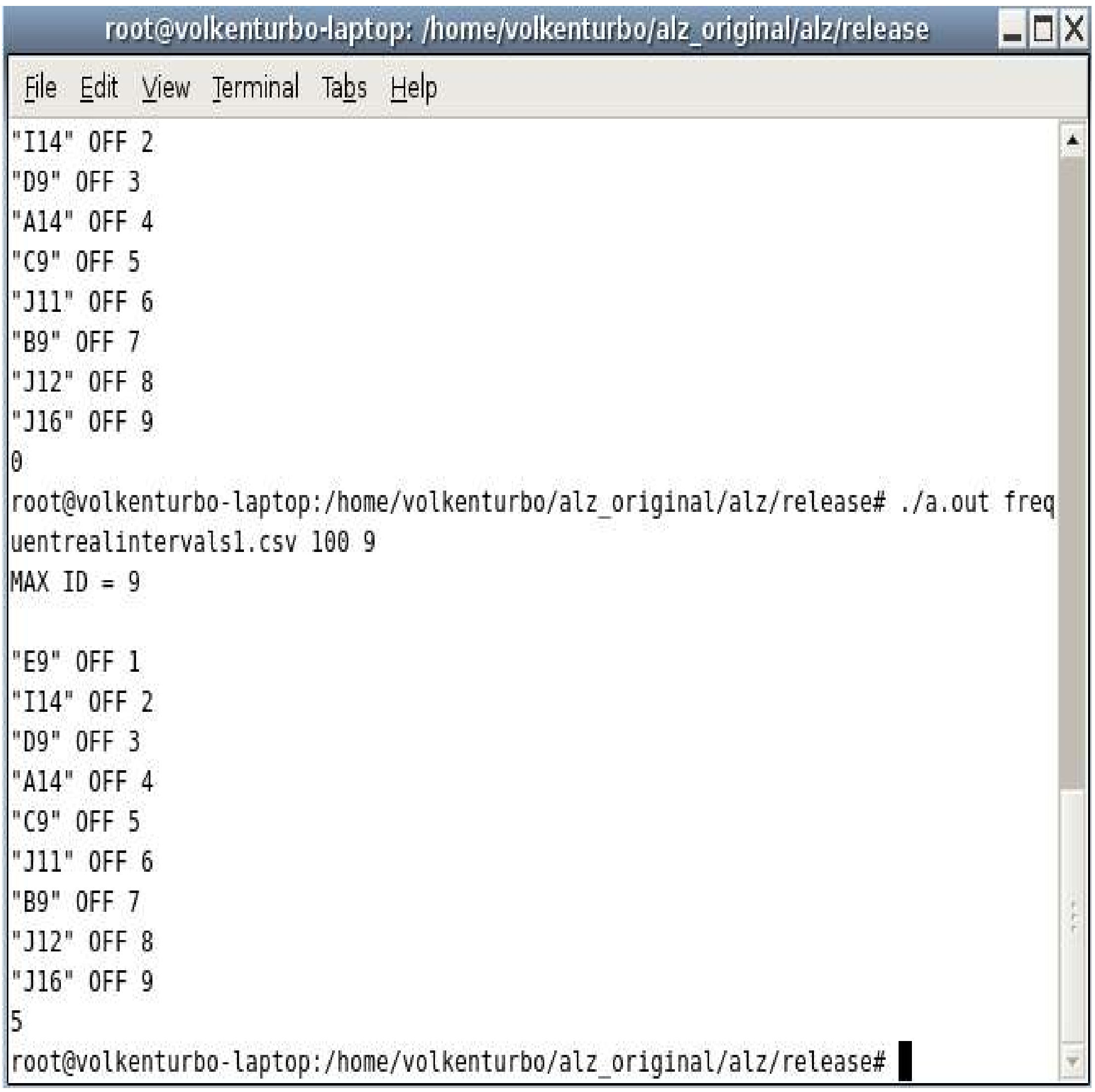


Figure B.10. Original sequential prediction output. (Note we pass the dataset and mention the training and testing samples and output the number of correct predictions.)

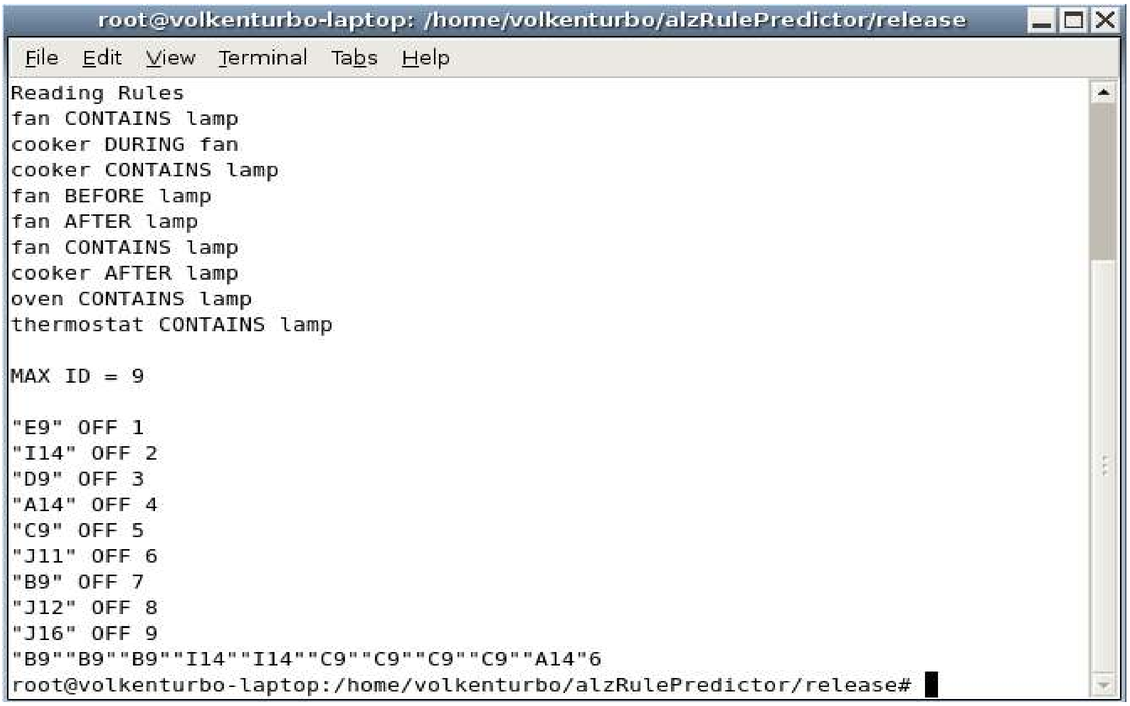


Figure B.11. Rule based prediction output.

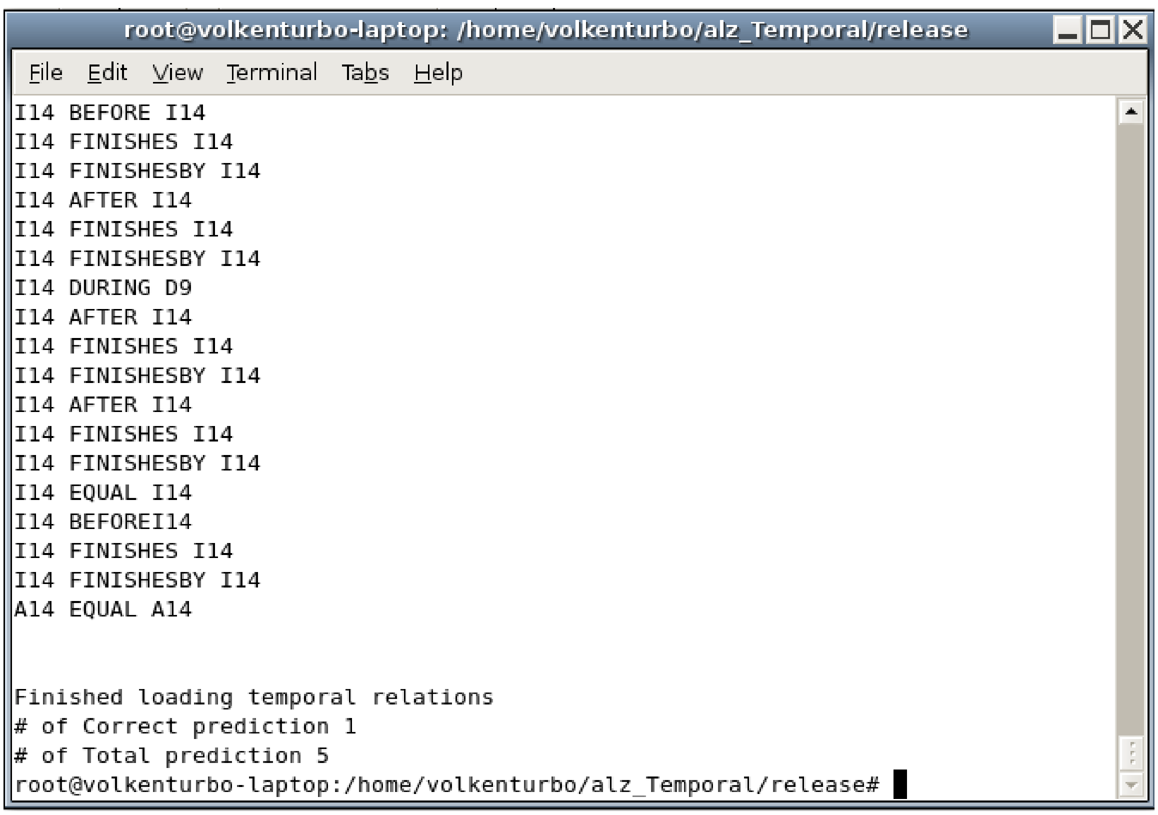


Figure B.12. Temporal relations based enhanced prediction screenshot.

TempAl: Visualization

The visualizer is a new interface for visualizing daily activities and this is now fitted with the pattern search option. Future work would include different pattern analysis options to get more interesting patterns which would help the resident to understand activity patterns in a home. In this visualizer we can notice the event by name as well as the start and end times, which are grouped by the date they occur. These event time intervals are represented by a line representation similar to a Gantt chart representation and help us visually examine the event time intervals.

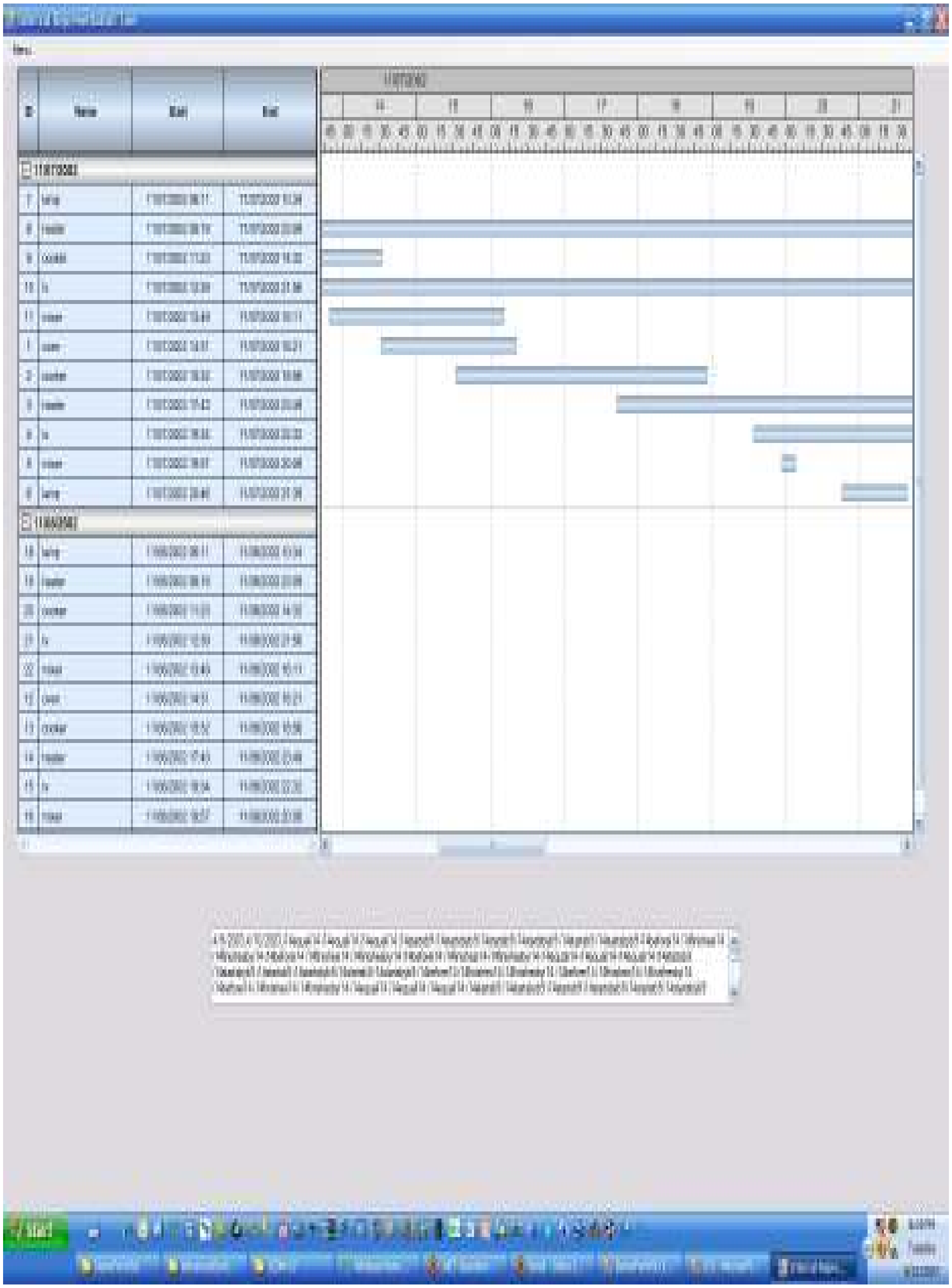


Figure B.13. Temporal relation visualization screenshot.

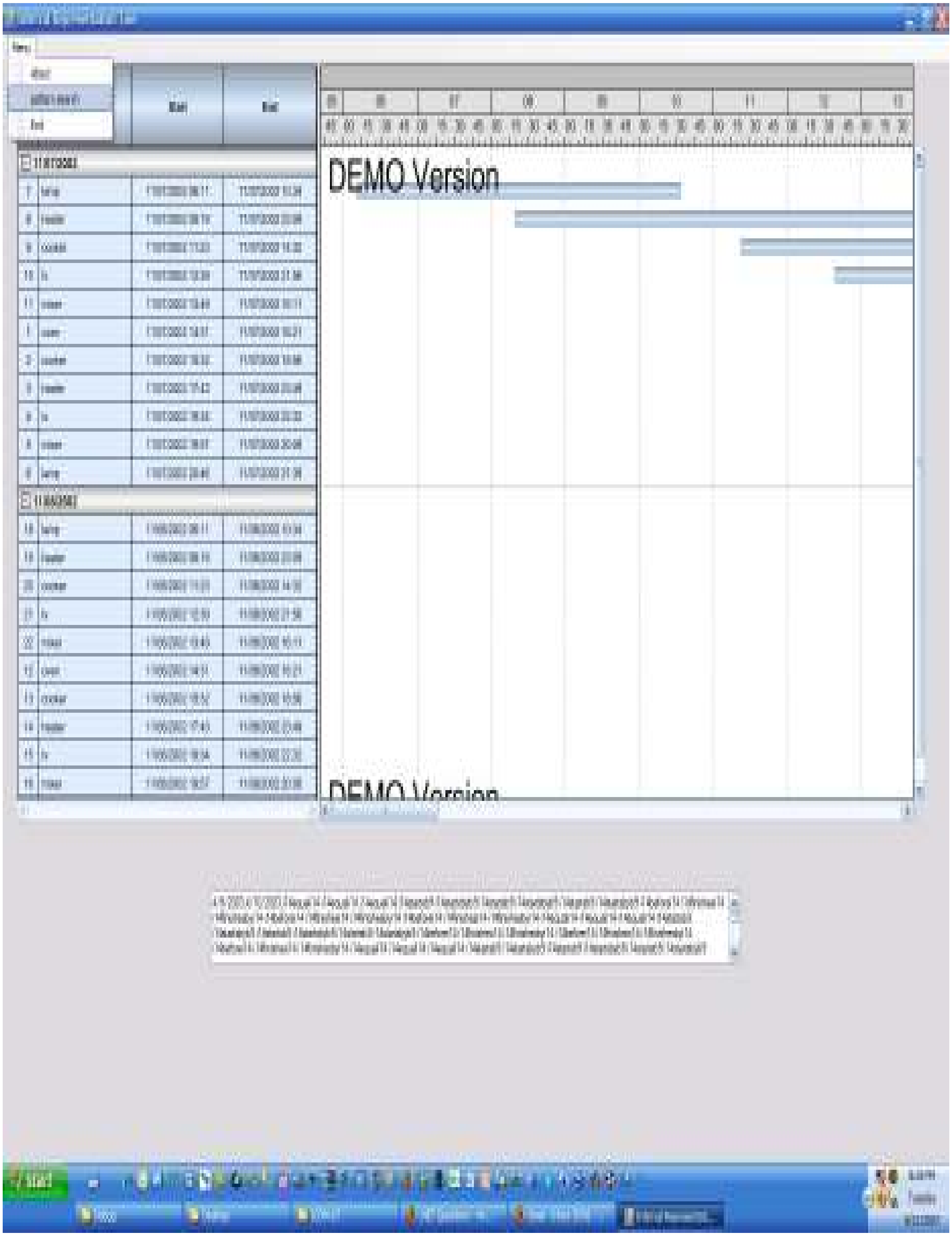


Figure B.14. Temporal relation visualization screenshot.

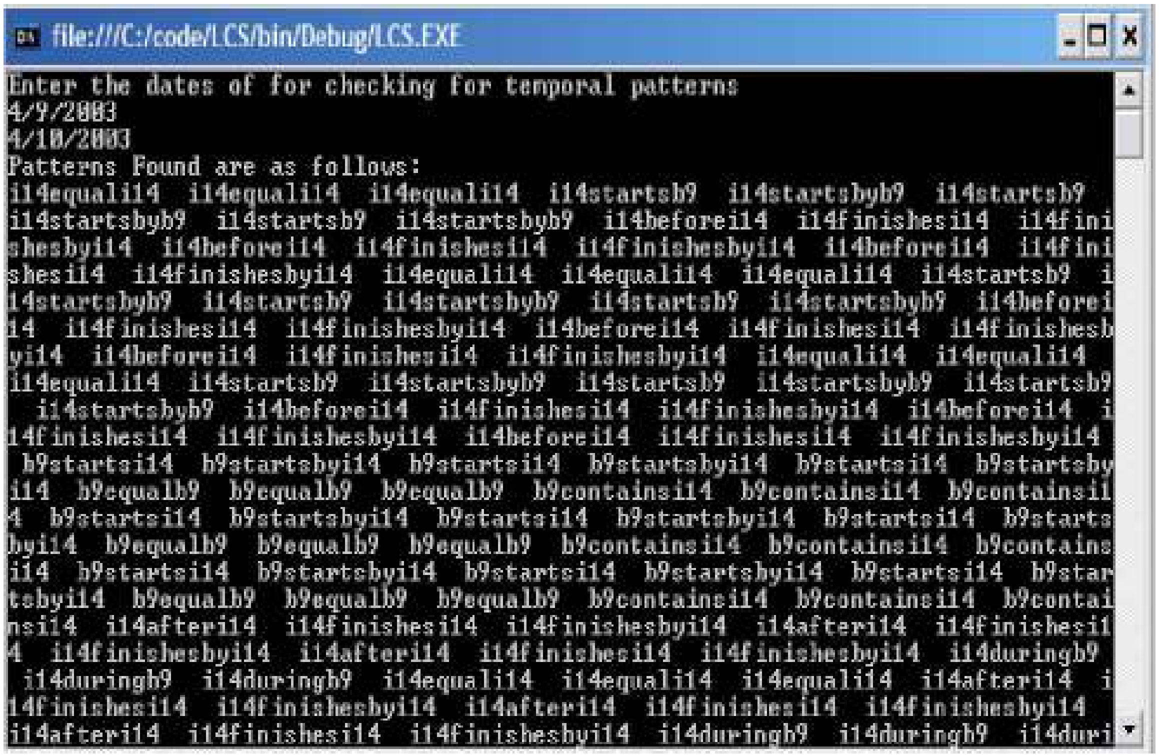


Figure B.15. Temporal relation LCS pattern search screenshot.

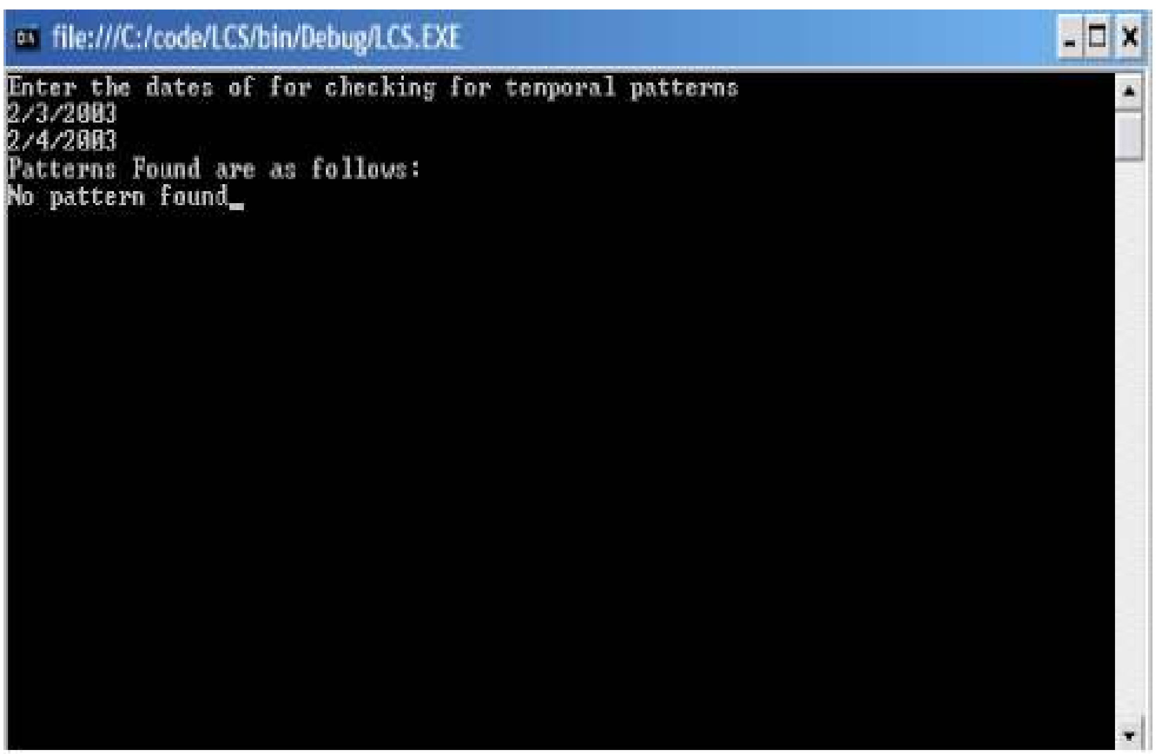


Figure B.16. Temporal relation LCS pattern search screenshot where no pattern is found.

APPENDIX C

ADDITIONAL TESTCASES

Test Cases with Varying Attributes

We perform statistical tests on the results obtained and find that the prediction algorithm with TempAl performs better than Alz but not significantly better or does not out perform. We note this would lead to additional testing by varying the following attributes which include number of different devices or events, number of anomalies present, number of patterns and number of instances available for training and testing. The observed outputs are displayed in Table C.1.

Table C.1 Final Test Case.

Attributes							Alz	Alz+ TempAl	Alz	Alz+ TempAl
#	No of Instances	Different Events	No of Anomalies	No of Pattern	Train Days	Test Days	# Correct	# Correct	Accuracy %	Accuracy %
1	6000	25	5	20	5000	1000	603	582	60.3	88.2
2	6000	10	15	20	5000	1000	411	425	41.1	42.5
3	6000	10	5	50	5000	1000	757	763	75.7	76.3
4	6000	10	5	20	5000	1000	913	914	91.3	91.4
5	3250	10	5	20	2500	750	668	716	89.06	95.4

We observe that Alz+TempAl would under perform when there is more number of events and less number of patterns as we see that the given patterns would not be sufficient for better analysis. Also, we see that while computing the probability of occurrence the more the number of distinct events lead to more temporal relations and the frequency of frequent relations decreases as newer and newer relations do come into existence and we see that the probability is measured by frequency counts of predictive temporal relations.

Alz + TempAl would perform better than Alz when more anomalies are included due to the fact that the temporal information is more than mere sequential probability, but the overall accuracy of prediction for both predictors is considerably low.

More number of scenarios would not affect both the learners and they perform considerable well. But with more training examples the learning increases and they start doing a good job by predicting better.

Alz+TempAl would perform better than Alz if provided with fewer instances to learn or in a fewer instance scenario. We see that thought both the methods uses frequency as a means to measure we observe that the temporal relations based TempAl + Alz would have more information and would quickly learn compared to Alz which would require more instances to learn better. Thus we see that the Alz + TempAl would be beneficial on smart home datasets.