IDENTIFYING CRITICAL EVENTS IN SMART ENVIRONMENTS USING ANOMALY DETECTION

By

VIKRAMADITYA REDDY JAKKULA

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

DOCTOR OF PHILOSOPHY

WASHINGTON STATE UNIVERSITY
Department of Electrical Engineering and Computer Science.

December 2016

To the Faculty of Washington State University:

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

_____
Committee Chair Full Name, Ph.D., Chair

_____
Committee Member Full Name, Ph.D.

_____
Committee Member Full Name, Ph.D.

_____
Committee Member Full Name, Ph.D.

_____
Committee Member Full Name, Ph.D.

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 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 my faculty at Washington State University 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 at Washington State University AI & ML Labs, for their interactions and discussions, as I believe they were a constant source of feedback for my progressing research. This work was supported by NSF grant.

`

Abstract

By Vikramaditya Reddy Jakkula, Ph.D.
Washington State University
Dec 2016

Chair: Dr. Diane J. Cook.

The need to have a secure lifestyle at home is in demand more than ever. Today's home is more than just four walls and a roof. Technology at home is on the rise and the place for smart home solutions is growing. One of the major concerns for smart home systems is the capability of adapting to the user. Personalizing the behavior of the home may provide improved comfort, control, and safety. One of the challenges of this goal is tackling anomalous events or actions. This work proposes multi-one class support vector machine learning techniques to address this issue of detecting anomalous events or actions in the smart environment datasets. The approaches are validated and discussed using real-world sensor data captured from smart home test bed. The critical event in data collected in a smart environment provides us with a better understanding of patterns that occur over time. Our pattern discovery and pattern matching using anomaly detection, 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 signatures of anomalous events, we can track critical events, the results can be used to enhance prediction and to detect critical events to warn the inhabitants or caregivers.

TABLE OF CONTENTS

Page

Dedication

Pranitha Reddy & Masi Reddy

CHAPTER ONE: INTRODUCTION

The smart homes are built by adding intelligent and adaptive behavior to home automation systems. This additional capability gives the user of smart home new tools to sense and adapt to their personal needs. As the population continues to age, providing tools to maintain independent living and support the aging in place philosophy to healthcare becomes ever more important. Smart home tools are also geared to address the increased cost of healthcare by reducing the load on care providers while finding ways to prevent medical emergencies. Given the costs of nursing home care and the importance individuals place on remaining in their current residence as long as possible, use of technology to enable individuals with cognitive or physical limitations to remain in their homes longer should be more cost effective and promote a better quality of life. A range of intelligent systems built for providing healthcare and wellness enables people to live at home with an improved overall quality of life [67].

A notable challenge to the deployment of these systems is designing anomaly detection algorithms which can improve existing techniques by identifying, and possibly filtering, rare and unexpected events. Detection of unusual events is an important issue in smart home research. However, this is a challenging task when designing an effective and computationally reasonable solution. This work demonstrates tools aimed at building a solution that detects abnormal behavior in sensor data collected in a smart home.

Anomaly detection is a set of techniques that are capable of identifying rare (anomalous) events in large datasets. Classical approaches to detection of unexpected events utilize a set of expert-defined rules to detect anomalous events. Anomaly detection has grown beyond simple rules by including statistical analysis and advanced machine learning techniques.

Anomaly detection offers many benefits to smart home research, as summarized in Table 1. The most common ones include identifying rare, unexpected events which may indication a situation of concern or interest. Additionally, filtering such events helps to improve learning algorithms, such as activity recognition, by reducing noise in the dataset. This process will also benefit adapting the smart home to the resident.

Table 1. Benefits of Anomaly Detection to Smart Sensor Data

• Standardization of Smart Sensor Datasets • Feedback to Learning Models • Promote difference between standard data and raw data • Reminder systems and prompting system performance improvement • Evaluation of human lifestyles & improvement suggestions

Anomaly detection also plays a major role in reminder systems. Filtering anomalous events will improve a prompting system's performance. Additionally, when anomalous events are noted the user may be informed of the unexpected situation through either audio, video or message prompts. Imagine a prompting system which identifies anomalous activity and provides prompts to take correct action when required.

The need for a robust anomaly detection model is 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. For example, if we are monitoring the well being of an individual in a smart home and the individual has not opened the refrigerator all day as they normally do, this should be reported to the individual and the caregiver and would be tagged as a critical event. 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.

Anomalous data detection has unique facets and possesses a number of challenges. Out of the various open issues to address when attempting anomalous event detection, this work focuses on the problem of whether a given sensor event is anomalous in nature by using a single class support vector machine. To validate this approach, experimental data was collected from real world settings with human subject participants. The experiment conducted is detailed in the experiment evaluation section and the results observed are presented. We believe that this approach to anomaly detection performs well and should enable smarter living in homes

CHAPTER TWO: CURRENT RESEARCH TRENDS

Recent advancements in multiple technology domains have positioned smart environments as feasible tools for assisted living, work spaces and other living spaces. This has been accomplished by advancing sensor technology, artificial intelligence, data mining, and machine learning techniques. Today, smart environments are equipped with a wide variety of sensors including motion, temperature, pressure sensors, and other intrusive/non-intrusive sensors, that allow the system to collect data on inhabitant activities and environmental situations and to later use them for automating the home. 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].

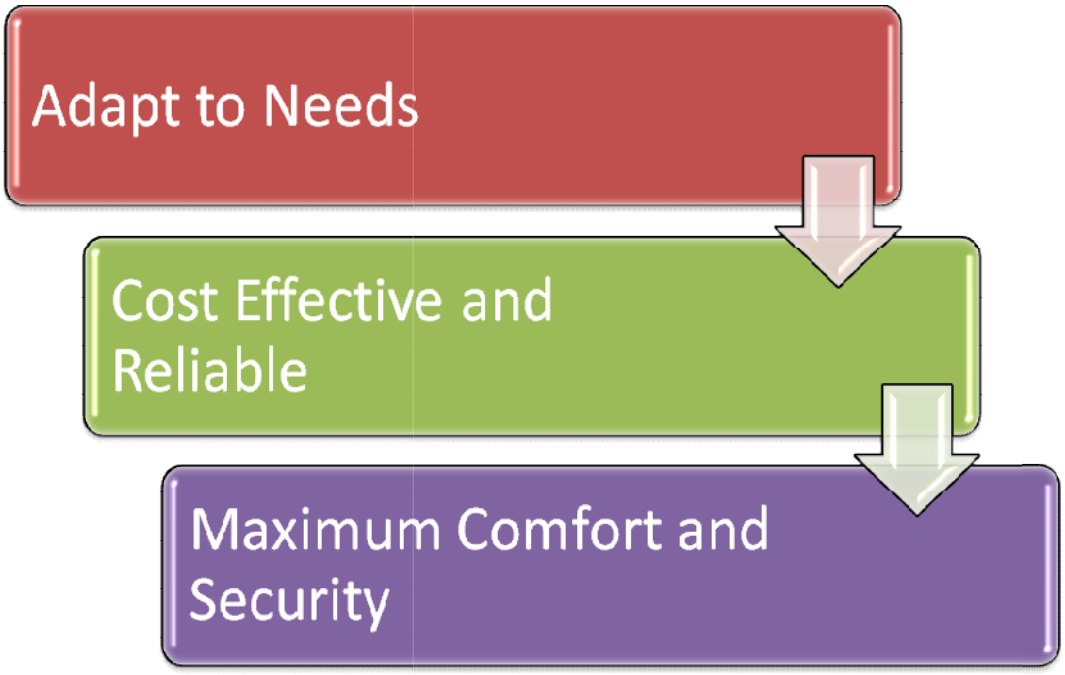


Figure 1. Common goals of the smart environment.

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 [7] [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.

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 2. 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.

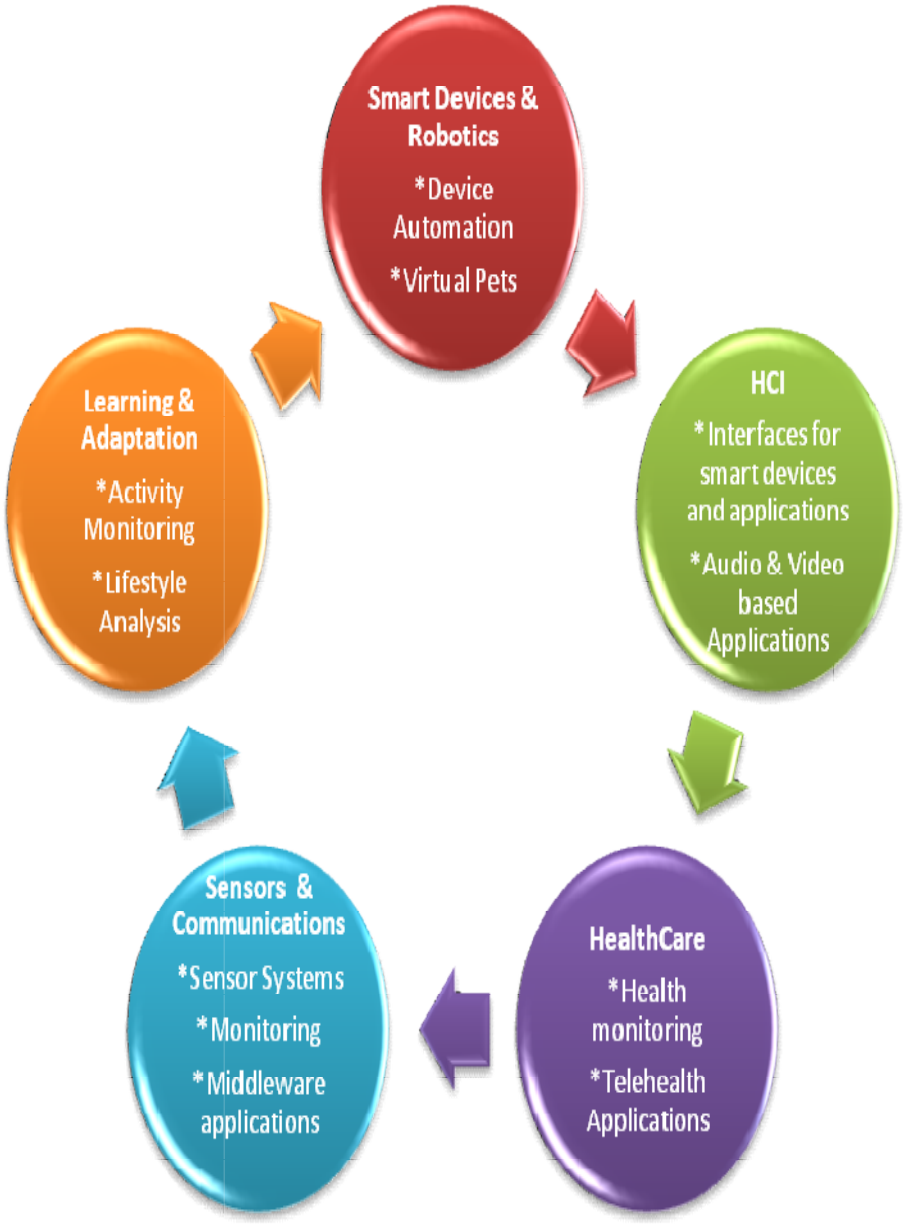


Figure 2. Current challenges in smart environments

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 2, illustrates some significant current projects being pursued in the research world.

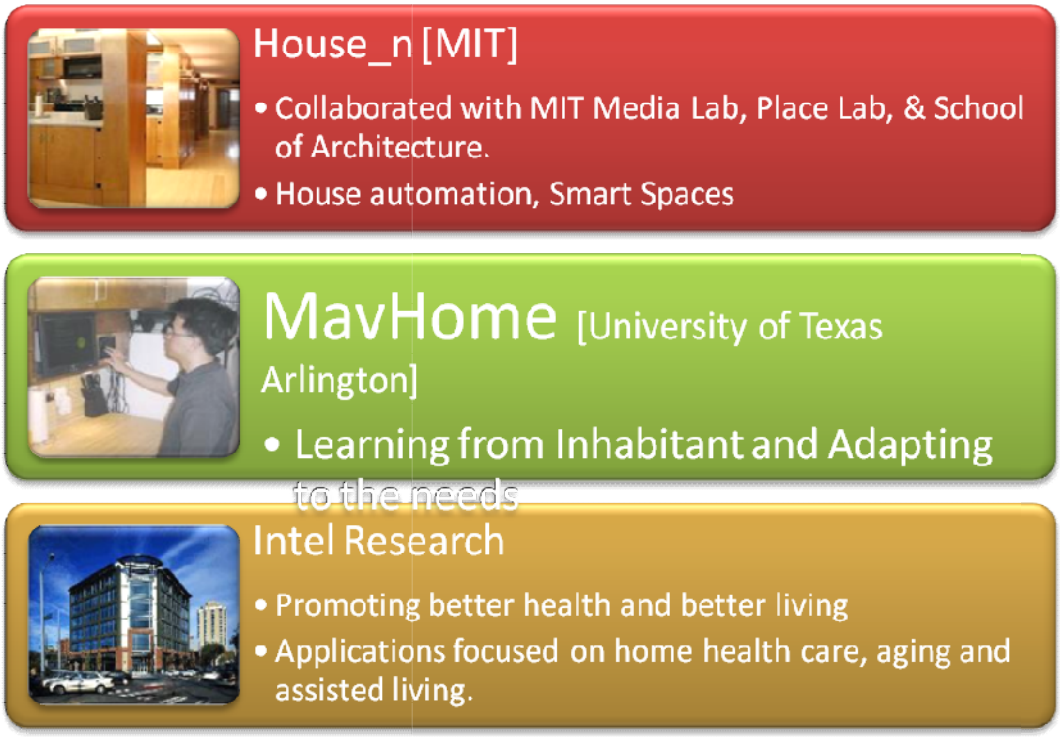


Figure 2. Current trends in smart environment and intelligent systems research at MIT House_n project, MavHome project and Intel Research [9] [10] [11].

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].

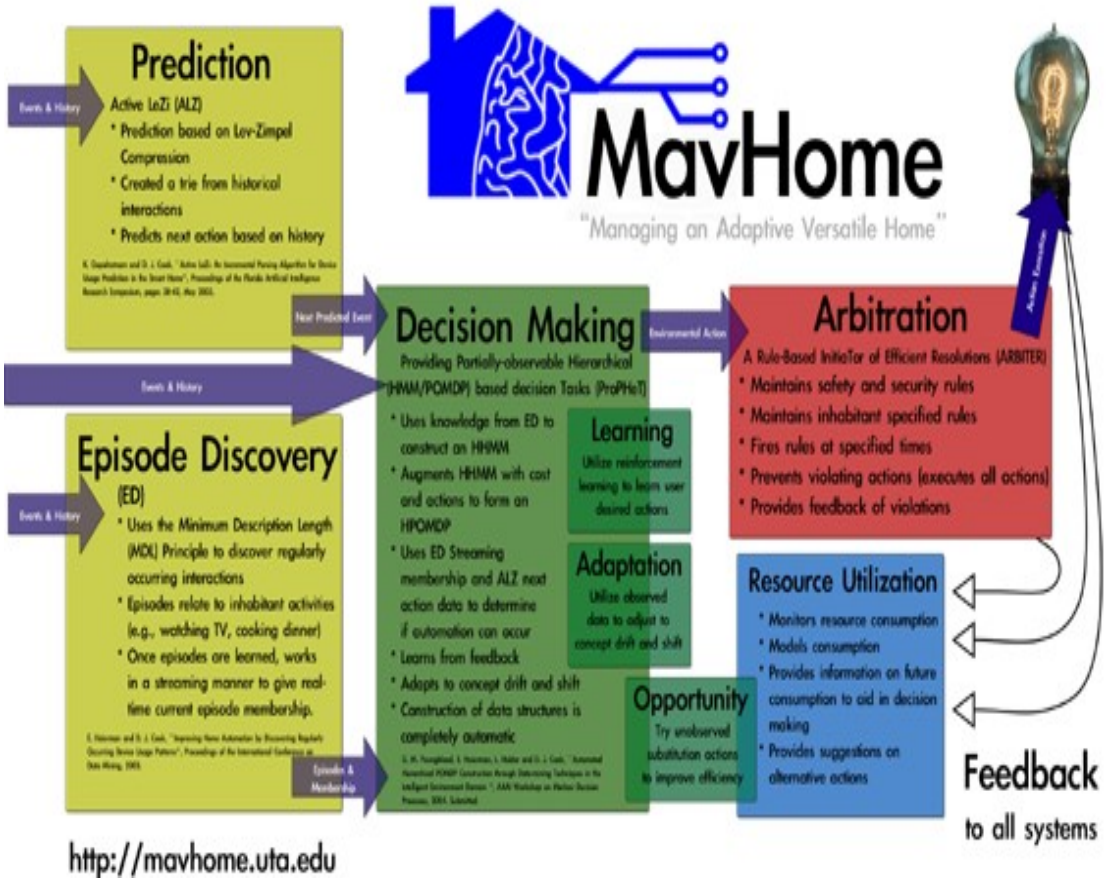


Figure 3. MavHome software architecture [8].

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.

Anomaly Detection

Anomaly detection is a relatively new field which is currently being approached and explored in smart home research. Some past approaches include a temporal-based approach where temporal relations [73] identified and probabilistic models were built to evaluate and identify anomalies [75]. An RFID-based approach was also experimented with, for human

behavior modeling and anomaly detection for elderly care. This approach presented a system for RFID data collection and preprocessing, clustering for anomaly detection, and promising experimental results [74]. Neural network-based approaches are also investigated where predicted values are used to inform the caregiver when anomalous behavior is predicted in the near future [76]. Anomaly detection is used for other domains too. Additionally, there are conceptual studies done and use cases reported for abnormal events in smart environment context.

Anomaly detection is the task of finding anomalous sensor data in datasets, whose behavior does not conform to the normal behavior of the majority of the data and these can be tagged as critical event in smart environments. These anomalous data have been of increasing interest in many domains, because their presence could indicate an unauthorized access of a system, credit card fraud, and failure in a part of a monitored system or a diagnosis of an unknown disease. There are many attempts to solve the anomaly detection problem. The approaches that are more widely applicable are unsupervised algorithms as they do not need labeled training data meeting the requirements of practical systems. The most commonly used unsupervised anomaly detection algorithms are K Nearest Neighbors based or Clustering based. In contrast to supervised machine learning, there is up to now no freely available toolkit such as the presented extension in the anomaly detection domain. Now, non-experts can easily integrate the implemented operators into complex processes using the intuitive graphical user interface of current generation tools with these implementations.

Anomaly detection algorithms could either be global or local. Global approaches refer to the techniques in which the anomaly score is assigned to each instance with respect to the entire dataset. On the other hand, the anomaly score of local approaches represent the outlierness of the

data point with respect to its direct neighborhood. The local approaches can detect outliers that are ignored using global approaches, especially in case of varying densities within a dataset. The anomaly detection extension contains two categories of approaches: nearest-neighbor based and clustering based algorithms. Algorithms in the first category assume that outliers lie in sparse neighborhoods and that they are distant from their nearest neighbors. The second category operates on the output of clustering algorithms being thus much faster in general.

Nearest-neighbor based algorithms assign the anomaly score of data instances relative to their neighborhood. They assume that outliers are distant from their neighbors or that their neighborhood is sparse. The first assumption corresponds to k-NN which is a global anomaly detection approach, while the second assumption refers to local density based approaches.

Nearest-neighbor based algorithms:

1. k-NN Global Anomaly Score
2. Local Outlier Factor (LOF)
3. Connectivity based Outlier Factor (COF)
4. Local Outlier Probability (LoOP)
5. Influenced Outlierness (INFLO)
6. Local Correlation Integral (LOCI)

The process of arranging similar objects into groups is referred to as clustering. Clustering based anomaly detection techniques operate on the output of clustering algorithms, e.g. the well-known k-means algorithm. They assume that anomalous instances either lie in sparse and small clusters; far from their cluster centroid or that they are not assigned to any cluster at all. The initial step followed by these algorithms is to classify the clusters into small

and large clusters. The user has the choice to select whether this partitioning is implemented similar to what was proposed in using two parameters α and β or using a single parameter γ.

Clustering based algorithms:

1. Cluster based Local Outlier Factor (CBLOF)
2. Local Density Cluster based Outlier Factor (LDCOF) A variant of CBLOF.

Summary

In this chapter we discussed the current trends in smart environment research and the current trends in anomaly detection research. Smart homes are intelligent environments which are designed used to make the life of residents easier and aid them in everyday activities.

CHAPTER THREE: ENVIRONMENTAL SENSING

Anomaly detection refers to the problem of detecting patterns in data which do not conform to normal or expected behavior. These techniques are extensively used in various applications such as: health care, intrusion detection in security systems, fault detection in safety critical systems, etc.

A standard process used in anomaly detection comprises of two main steps. The first is defining a region that represents normal behavior and the second is making decision whether an observation belongs to the normal region or does not. Factors which make the decision process a complex task are:

- boundary between normal and anomalous behavior is often not precise;
- in many domains normal behavior keeps evolving, therefore the region dedicated as normal could change in the future;
- small deviation from normal behavior could be an anomaly in some domains, whereas in other domains they might be normal;
- labeled data for training model for anomaly recognition are often not available;
- data may often contains noise which is difficult to distinguish and remove;

When speaking about a smart home environment the big stress is given on the ”smartness” of such a system. However, the question is what is meant by the ubiquitous intelligence, how a smart home should react to a particular behavior of their inhabitants, etc.

Some approaches have addressed several questions regarding the abnormal behavior detection in their work:

What type of abnormal behavior may occur?

How does the system reason about its inhabitant's behavior?

How should it react to abnormal behavior?

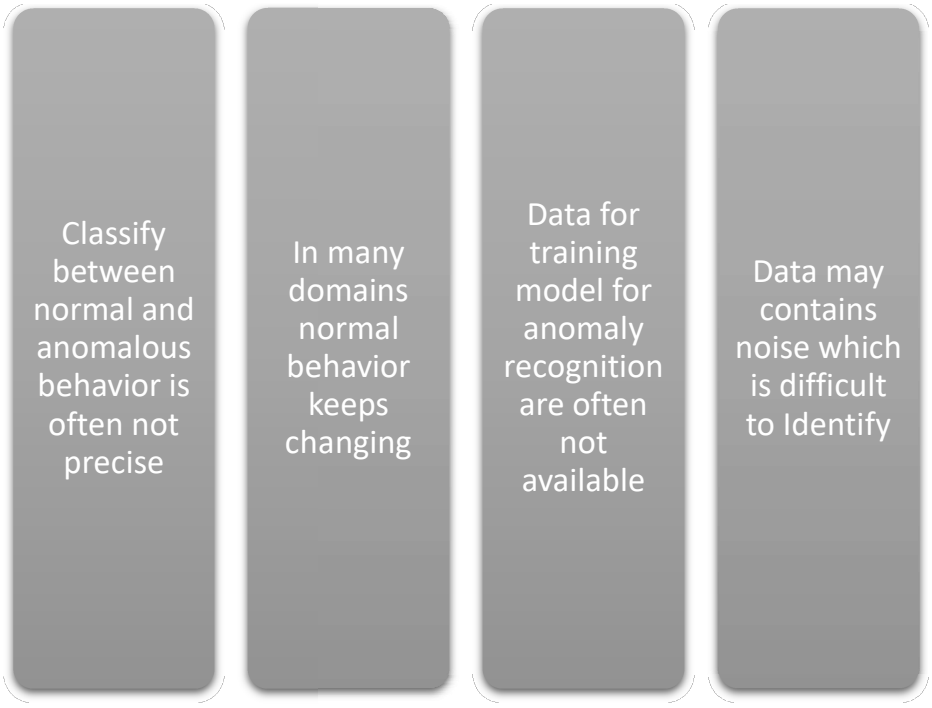
What information should be involved in abnormality?

To find answers to these questions it is needed to get along with accessible data, to understand user activities from these data and to provide adequate response. If the system will bother the caregiver every hour with some notification about what elderly is doing, the satisfaction with the system will obviously be very low. On the other hand, if the system will not provide an alert in every situation when elderly is in danger, it may have unwanted consequences.

Some use cases are:

- Abnormality in duration - abnormal activity takes longer or shorter as usual.
- Abnormality in time of occurrence - an inappropriate activity start time could imply illness or even dementia, but a forgetful person may only need a reminder.
- Inappropriate actions - the additional behavior does not conform to the norm. Some actions have been taken, which are usually not.
- Wrong places - performing an action in the wrong place (e.g., jumping on the bed or lying on the kitchen floor) may be dangerous or signify that something has gone wrong

Figure 4. Factors which make the anomaly detection process a challenge



A smart environment may be defined as a system that collects data about the inhabitants of a living space and the environment in order to model and adapt the environment. This allows the space to adapt 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. The history of observed sensor events reflects activities that occur in the environment and can be used to discover frequent recurring activity patterns, to recognize activities of daily living, identify suspicious states, and to predict resident actions.

The data used for this work's experimentation was collected from real smart home test beds in the CASAS environment, with more details available in experimentation section

below. The data was later stored into a database and later annotated by a human to provide a ground truth.

The sensor data collected by this system is expressed by several features, as summarized and illustrated in Table 2. There are five fields represented as follows include Date, Time, Sensor ID, Message and Annotation.

Table 2. An example for data collected from smart environment

Date – Time – Sensor – Message – Annotation – Annotation State
2009-06-15 17:07:52.312001 D031 OPEN Enter_Home begin
2009-06-15 17:07:54.921001 M006 ON
2009-06-15 17:07:58.828001 M006 OFF
2009-06-15 17:08:00.218001 M015 ON
2009-06-15 17:08:00.562001 D031 CLOSE Enter_Home end
2009-06-15 17:08:04.515001 M015 OFF

The data for the experiment is collected from three different test bed set ups with real residents and the activity is recorded as mentioned above. The resulting possesses millions of data points.

Also to note is that only non-intrusive sensors were used to collect the data. The goal of these systems is to be as non-intrusive as possible and by using passive, low profile sensors the smart home is designed to allow the residents to live in their home as normally as possible.

The test beds used for this work contained a living room, dining area and kitchen. The activity level for each of the smart test beds is illustrated by Figures 5. The illustration presents activity occurrences, activity density, sensor frequency distribution, and activity time distribution for the test beds B1 which is part of the experiment.

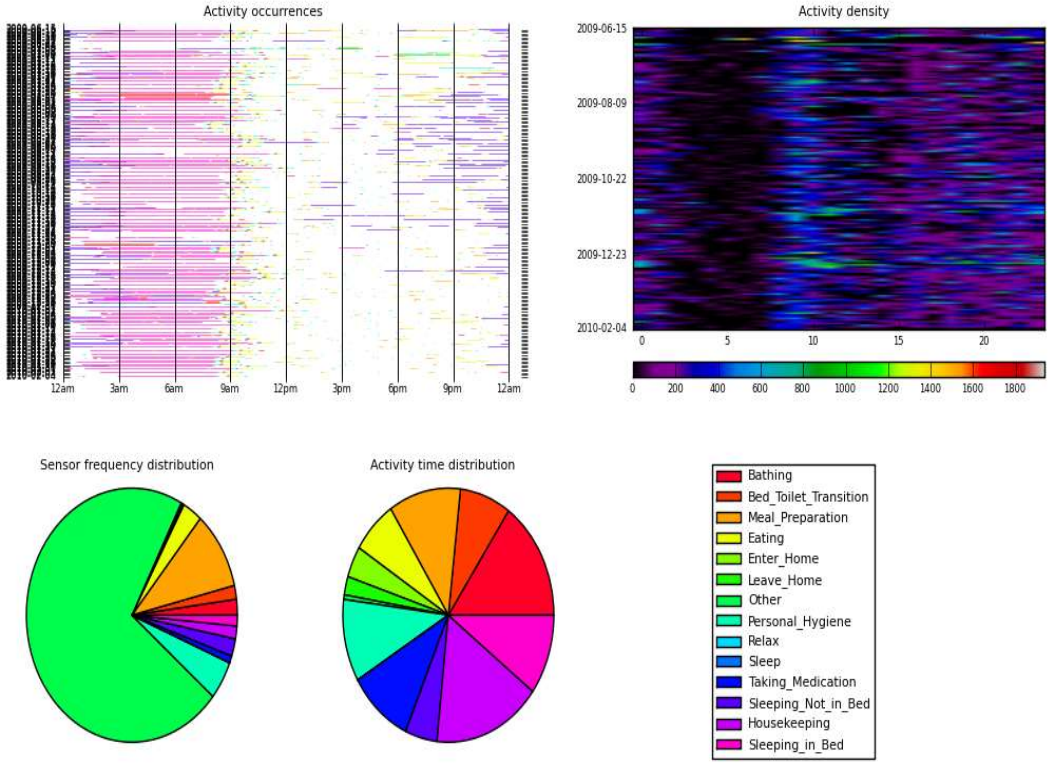


Figure 5. Smart test bed Code name "B1" activity pattern

While mining datasets, we often encounter the problem of noise and outliers. The meaning of outliers was already described, while the noise complicates the shape of the decision border of the prediction system.

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. 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 370,000 actions representing six months of activities. The synthetic data consist of 30+ devices which can be considered to be spread across a virtual environment of five rooms and simulates these device

activities for a 180 day period. During the generation of synthetic data we inject anomalies based on timed scenarios.

Example of Synthetically injected scenarios: Un expect and Rare event

1) M005 -- Shower activity in night

2) M002 -- Cooking in night,

3) M010,M011 -- Dining in the night

4) M003 -- Sleeping during afternoon

5) m020 -- Workspace in the night

6) m009 -- hALL WAY IN NIGHT

7) m006 - enter/leave home in night or when supposed to sleep regularly

Example of anomaly detection synthetic data

2009-08-11 11:56:57.453001 M015 ON Anomaly begin

2009-08-11 11:56:58.218001 M014 OFF

2009-08-11 11:57:00.031001 M001 OFF

2009-08-11 11:57:00.562001 D021 OPEN Meal Preparation begin

2009-08-11 11:57:01.515001 M015 OFF Anomaly end

2009-08-11 11:57:02.328001 M002 ON

2009-08-11 11:57:03.015001 M001 ON

Performance of computer and sensor systems is commonly characterized by the occurrence of rare events which can be termed as critical events such as falling from stairs and laying motionless, not

watching television for the entire day and so forth. Straightforward simulation for this type of rare event leads to simulation run times in the order of months and years, thus requiring new methods to be investigated and employed for generating large datasets with some solid ground truth of rare events or anomalous events. An extremely useful feature of the Monte Carlo method is that it can be readily modified for finding (estimating) the optimal solution in an NP-hard combinatorial problem.

Summary

In this chapter we introduced environment sensing for data collection. We also gave a description of the environment in which we collected data for our algorithm validation. We have used real world data as well as artificial dataset.

These outstanding patterns are also known as anomalies, outliers, intrusions, exceptions, misuses, or fraud. The name usually refers to a specific application domain, thus, we are using the generic term anomaly in the following. Anomaly detection can basically be classified as a sub-area of data mining and machine learning. The first attempts at anomaly detection go back to the 1970s, where researchers attempted to remove noise or incorrect measurements from their data in order to ensure that the data fits best to their proposed models. In this context, the Grubbs' outlier test [1] is most likely the best-known algorithm. It takes univariate (one-dimensional) input data assuming it is normally distributed and determines whether the minimum and maximum of the data is an outlier based on a significance level α. Once an outlier has been found, it is removed from the data and the procedure is repeated. It is obvious to see the shortcomings of this algorithm: it works only on univariate data and it fails if the data is not normally distributed. However, it illustrates the first attempts at removing anomalies, also known as data cleansing. a more modern approach of anomaly detection has become increasingly more popular in the last number of years and has attracted a growing and active research community, which began in 2000, approximately. This was attributed to the fact that focus changed; in particular, people were not primarily interested in data cleansing any longer but in detecting anomalies instead. The rapid growth of databases and the availability of massive log data led to the desire to discover anomalous records, deviating from the norm. In addition, the output of such algorithms evolved. Instead of having a binary label indicating an anomaly, algorithms which assign scores to all instances indicating their extent of being an outlier, gained attention

Today, anomaly detection is applied in many application domains. In the area of network security, it is used in network intrusion detection systems [2]. Suspicious behavior of malicious software, unusual network activity, and break-in attempts should be detected without prior knowledge of the specific incident. In modern anti-virus applications, behavioral analysis also complements the traditional pattern matching detection engines [3]. In addition, this example describes the difference of traditional

classification with the prior knowledge of the classes (malware patterns) and anomaly detection, dealing with the question if there is new previously unseen activity. In data leakage prevention (DLP) systems, researchers also try to incorporate anomaly detection algorithms for detecting abnormal data access patterns in order to avoid data to be stolen. A second core application area is misuse detection in transactional data. In this context, anomaly detection is used to detect fraudulent credit card transactions caused by stolen credit cards [4], fraud in Internet payments, or suspicious transactions in financial accounting data [5]. In the medical domain, anomaly detection is also used, for example, for detecting tumors in medical images or monitoring patient data (electrocardiogram) to get early warnings in case of life-threatening situations [6]. Furthermore, a variety of other specific applications exist such as anomaly detection in surveillance camera data [7], fault detection in complex systems, or detecting forgeries in the document forensics domain [8]. If anomaly detection should be applied on image data as shown in some of the above examples, representative features need to be extracted from that data first. This is a non-trivial task and, in some cases, for example document analysis, the pre-processing evolved to its own research domain.

Point Anomalies

Point anomalies are the simplest type of anomalies. A single instance differs from the others according to their attribute values. If the dataset is visualized, these anomalies can be seen immediately, at least when not having more than three dimensions (attributes). In Figure 23.2, the points p1 and p2 are such anomalies, having a large distance to their neighbors. Nearly all anomaly detection algorithms detect this kind of anomaly, which is why they are sometimes called point anomaly detection algorithms. In the remainder of this chapter, all presented algorithms are of this type. The above example which demonstrated the sales prices of products and their profit is also an example of a point anomaly detection problem.

Contextual Anomalies

Contextual anomalies can only be identified with respect to a specific context in which they are appearing. Since this is best explained using an example, please have a look at Figure 23.3, which shows the average monthly temperatures in Germany over the years 2001–2010. If a point anomaly detection algorithm was directly applied on the univariate temperature values, only the extreme temperatures could be found, which might be the peak in July 2006 or the extreme low temperatures in December 2010. However, it is obvious to a person that the mild winter in December 2006/ January 2007 is an anomaly with respect to the context time. In order to detect such contextual anomalies with point anomaly detection algorithms, the context needs to be integrated when generating an appropriate data view.

Collective Anomalies

If a certain combination of instances defines an anomaly and not only a single instance, we have a collective anomaly detection problem. A good example for this type of anomalies is a network intrusion detection system (IDS). Imagine an IDS logs all system calls of a computer.

Supervised Anomaly Detection

Supervised anomaly detection is a synonym for supervised learning. In this mode, a labeled training set is used to train a classifier, which is afterwards applied on a (labeled) test set. The only difference between supervised anomaly detection and supervised machine learning is the fact that the prior of the anomalous class is low (times less outliers compared with the normal instances). In general, any supervised learning algorithm, such as Support Vector Machines, Neural Networks, Decision Trees, or k-nearest-neighbors can be used in this mode. However, this scenario is extremely rare in practice since the anomalies are in most cases unknown in advance.

Semi-Supervised Anomaly Detection

Semi-supervised anomaly detection is also divided into training and a test phase. In contrast to supervised anomaly detection, the training dataset does not contain any anomalies but only examples of the normal class. During the training phase, a model for the “normal” behavior is learned and afterwards evaluated using the test set, which then contains normal Anomaly Detection 417 records and anomalies. A showcase of this mode is the network intrusion detection use case. In this context, normal behavior of a system can be learned, but the attack patterns (the anomalies) are unknown in advance. During the operation of the IDS, unusual situations can then be reported as suspicious activity to an administrator.

Unsupervised Anomaly Detection

Unsupervised anomaly detection is the most difficult mode. In this case, no assumption about the data is made and no training of a model is performed. The data is analyzed based on statistical measures only, for example, the distances to the nearest neighbors, and an anomaly score rates the single instances.

Anomaly Detection in Smart Homes

During the last twenty years, the ‘smart’ in ‘smart home’ has become more important [1, 2] with a focus on behaviour recognition and subsequent abnormal behaviour detection. There have been many proposed frameworks and algorithms for behaviour recognition [3], but little discussion of what precisely a smart home that monitors behaviour should do. Here we focus on smart homes for elderly care monitoring, discuss Use Cases that describe a situation, and then propose suitable outputs from the smart home.

Abnormality in Duration : Example An over-long shower

Goal: To detect an unusually extended activity. Initial state: Mary was at home alone. Description: Mary woke up at 0800. She began her morning shower at 0810, as usual, but at 0840 the motion sensor in the bathroom still indicated movement, and the shower tap was still on, so her shower had lasted for 30

minutes. Norm: Mary normally showers for 10 to 20 minutes. Outcome: An alert message was sent to Debbie, who called Carita. Carita discovered Mary confused and cold in the shower, having forgotten what she was doing. System design implications: An excessively long activity may put the smart home inhabitant at risk. This poses the following questions: When does a shower become longer than usual? 1 minute over the average? 5? 10? Or is the amount of overrun relative (1%? 5%? 10%)? Perhaps a more sophisticated statistic is appropriate (>1 standard deviation from the mean).

Anomaly in time of occurrence

Variation in shower start time Goal: To recognize acceptable variation in the start time of an activity. Initial state: Mary is home alone. Description: One cold winter morning, Mary awoke at 0800. She decided to wait until 0830 before taking her shower. The system noticed that Mary did not take a shower from 0800 to 0820 as had previously occurred, and generated a reminder for Mary and a warning message for her daughter, Debbie. Mary ignored the reminder, and waited until 0830 as she had intended. After work, Debbie checked the system and recognized that the system had made an incorrect inference that had occurred because it had not observed this activity in the winter. Debbie then provided feedback to the system to update this activity start time. Norm: Mary's shower starts in the time-range 0800 - 0820. Outcome: Mary's shower time was accepted at 0830. System design implications: In general, it is probably safe to assume that activity start times more than 1 standard deviation from the mean are interesting but not inherently problematic behaviors. Therefore it is acceptable to request external (human) input regarding the classification of the behavior, and it may not be necessary for the Smart Home to rely on pre-loaded world knowledge.

Performing activities in the wrong places

Use Case C1. Lying down in the kitchen Goal: To react to some abnormal behaviours immediately, as they are potentially very significant. Initial state: Mary was at home alone. Description: At 0815, Mary went to the kitchen to prepare her breakfast. She got some bread and put it in the toaster. She walked around the kitchen while she was waiting, and then suddenly lay down on the floor. The behaviour was

recognized, and then the spatial properties of the activity checked. As this behavior should not be seen in the kitchen, an alert was created, and marked urgent as the behavior was potentially serious. Norm: Inhabitants do not lie down in the kitchen. Outcome: An alert message was sent to both Debbie and Carita. System design implications: There is a class of activities that fall outside the bounds of normal behavior and can be prima facie assumed to be both interesting and problematic. This should reduce the computational effort involved in deciding how to react to an observed activity. However, complementing that is the difficulty of foreseeing all possible inappropriate behaviors. It's difficult enough to build a world model that allows for normal behaviors, but the size of the problem is potentially much larger if the system has to detect all possible dangerous abnormal behaviors.

Abnormality in a behavior pattern Making tea with sugar

<write explain example>

Current research is oriented on activity classification and techniques that determine whether observed behavior falls into predefined sets of activities. Other research is focused on finding anomalies in predefined scenarios [7]. However, both of them are limited to a small set of specific activities, which do not correspond to real behavior of the user at home. Techniques like automatic inactivity detection [12] algorithm lack some additional information about the context of the user. For example when the user goes to sleep late, it is assumed that he will sleep longer, though this information is not incorporated in this technique

Experiment 1: One Class SVM

The experiment done in this work consisted of parsing the data, training the learning algorithm and testing it against test data. After the testing data is passed through the classifier observations about its performance and capabilities are made.

The core component of this experiment is evaluating the one class support vector machine as an anomaly detection tool. This technique has been successfully applied in various domains and had good

results. The training data consists of annotated data while the testing data consists of annotation free data. Annotated data is clean data without anomalies and is cleared of any outliers by using Interquartile range filter available via the Weka tool. (Figure 6)

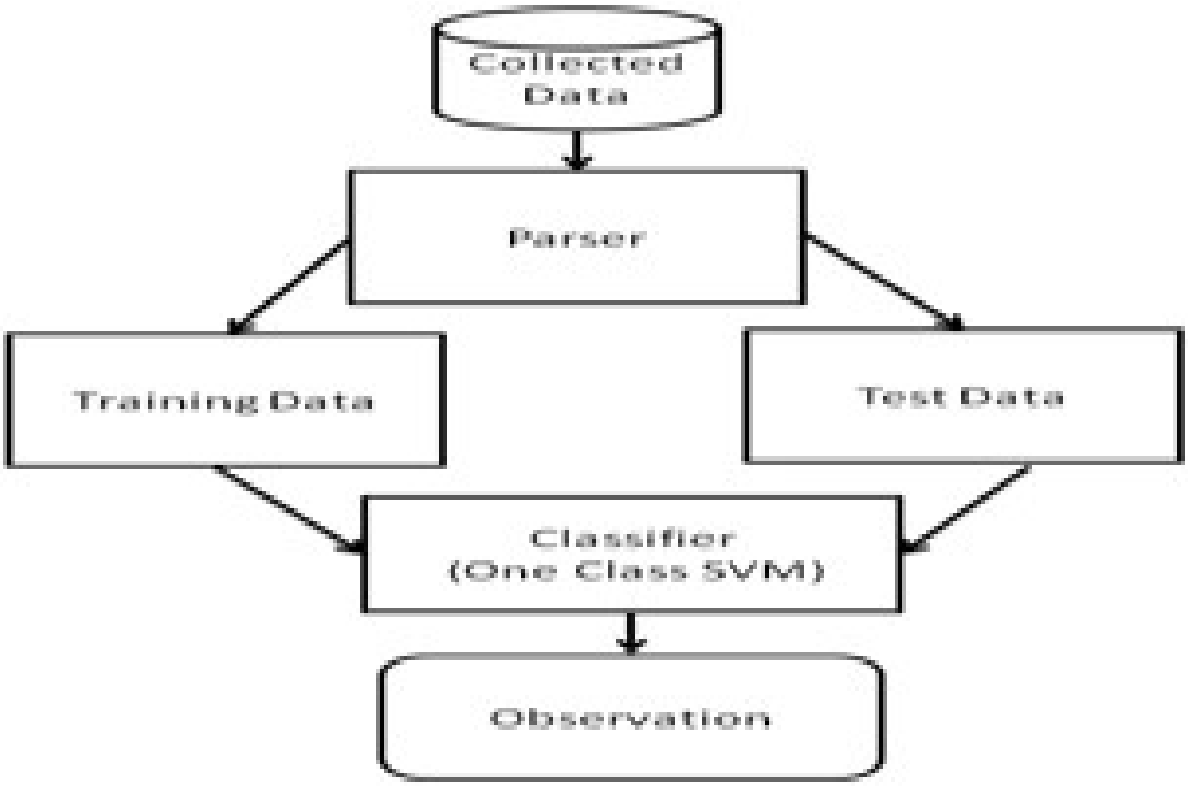


Figure 6. Experimentation Process

The test data consists of annotation free samples. The data used for the experimentation is created by a parser tool which improves the dimensionality of the dataset by introducing additional attributes. These attributes include the daily count of sensor occurrence for a particular day, monthly count of sensor occurrence for a particular month, and yearly count of sensor occurrence for a particular year.

One Class Support Vector Machines (OCSVM) are quite popular for anomaly detection problems. Suppose that a dataset has a probability distribution P in the feature space. The goal would be to find a "simple" subset S of the feature space such that the probability that a test point from P lies outside S and is bounded by some a priori specified value.

Supposing that there is a dataset drawn from an underlying probability distribution P, one needs to estimate a "simple" subset S of the input space such that the probability that a test point from P lies outside of S is bounded by some a prior specified v □ (0, 1) . The solution for this problem is obtained by estimating a function f which is positive on S and negative on the complement S. The algorithm can be summarized as mapping the data into a feature space H using an appropriate kernel function, and then trying to separate the mapped vectors from the origin with maximum margin (see Figure 3).

$$f(x) = \begin{cases} +1 & \text{if } x \in S \\ -1 & \text{if } x \in \overline{S} \end{cases}$$

In our context, let x1, x2, . . . , xn be training examples belonging to one class X, where X is a compact subset of RN. Let Φ : X → H be a kernel map which transforms the training examples to another space. Then, to separate the data set from the origin, ones need to solve the following quadratic programming problem :

$$min \frac{1}{2}\|w\|^2 + \frac{1}{vl}\sum_{i=1}^{l}\xi_i - \rho$$

subject to

$$(w \cdot \Phi(x_i)) \geq \rho - \xi_i \quad i = 1, 2, ..., l \quad \xi_i \geq 0$$

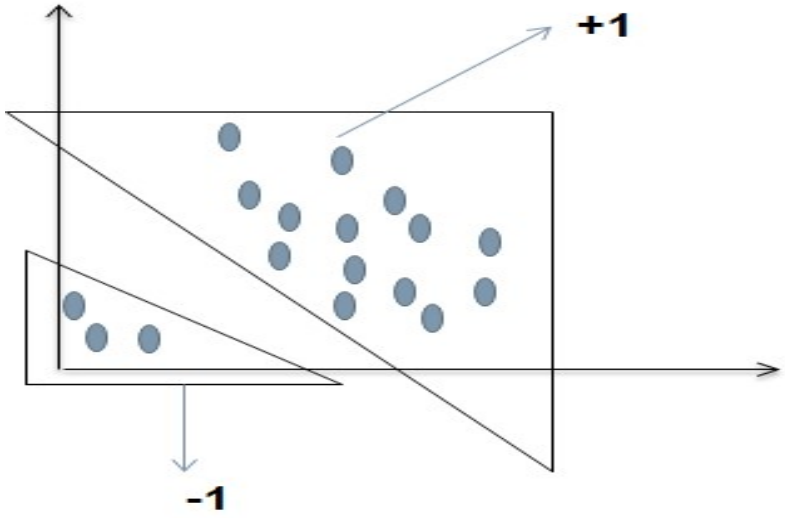


Figure 7. One Class Support Vector Machine

The LIBSVM algorithm available via Weka was used for this work. This is an integrated tool for support vector classification and regression which can handle one-class SVM using the Sholkopf algorithms. The standard parameters of the algorithm were used.

Cross validation was performed on the entire dataset after cleaning outliers or extremes. And report the observations as well as report the findings for train and test sets. We test the data with samples with no anomalies and report the findings. We assume the test set to be positive samples with no anomalies and report the observations. For the experimentation the RBF kernel with default parameters was used.

The evaluation metrics used for this experimentation include precision, recall and F-measure and type I, type II errors. Table 3 shows the observations on the test set run of the experiment.

Table 3. Experiment Observation on Test Set

		Test Set
B1	Type I	0.5
	Type II	-
	Precision	1
	Recall	1
	F- Measure	1
B2	Type I	0.4
	Type II	-
	Precision	1
	Recall	1
	F- Measure	1
B3	Type I	0.5
	Type II	-
	Precision	1
	Recall	1
	F- Measure	1

Given that the ground truth is not provided for testing, we train the SVM with positive samples and provide positive samples to test; hence type II errors are not considered. The results are presented in table 3, and lead way to the next steps where we increase the dimensionality and vary the kernels and hyper-parameters to observe the performance with ground truth-based anomalous datasets.

In future work, the plan is to extend this work by introducing multiple one class support vector machines. This would provide a SVM for each annotation to catch anomalies. This approach is a major step for anomaly detection in smart sensor datasets.

Experiment 2: Multi-One Class SVM

In this we use multiple one class classifiers to identify anomalies. It is an extension to the experiment above. To analyze the dataset we need to have a better picture of the data at hand. We need to look at the sensors and also the activity at home. We build a tool called AnaDet which can be used to check for anomaly detection and consist of one class and multi one class support vector machine algorithm. The below illustrations give an overview as how the tool looks and functions.

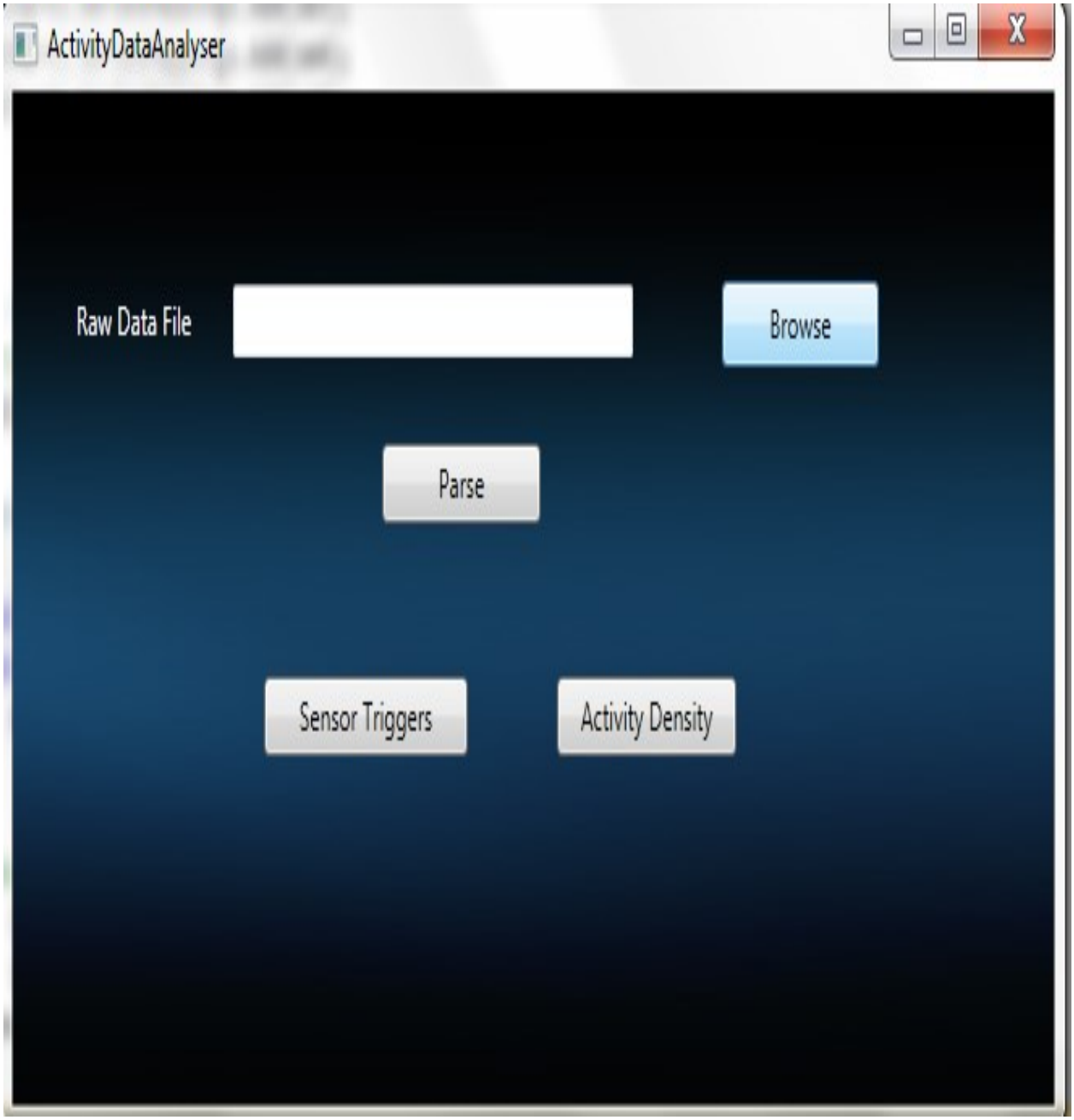


Figure 8. Activity Data Analyzer.

This tool is the activity data analyzer which can be used to parse the smart home datasets and create separate files for activity density monitoring and sensor triggers.

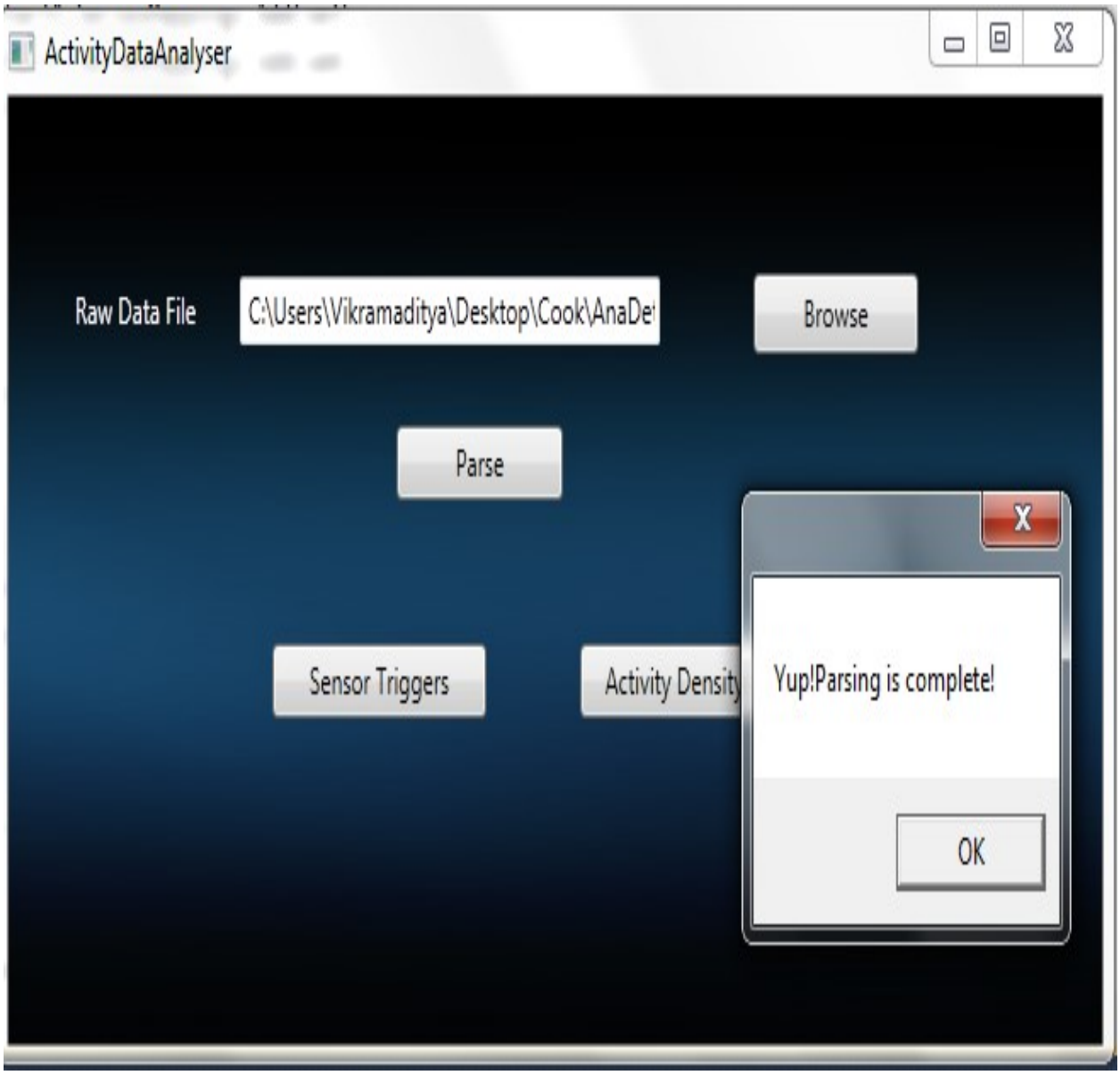


Figure 9. Result of parsing of Activity Data Analyzer.

The figure 9 displays that when the parsing is completed the results are saved into text files which are located in the local drive.

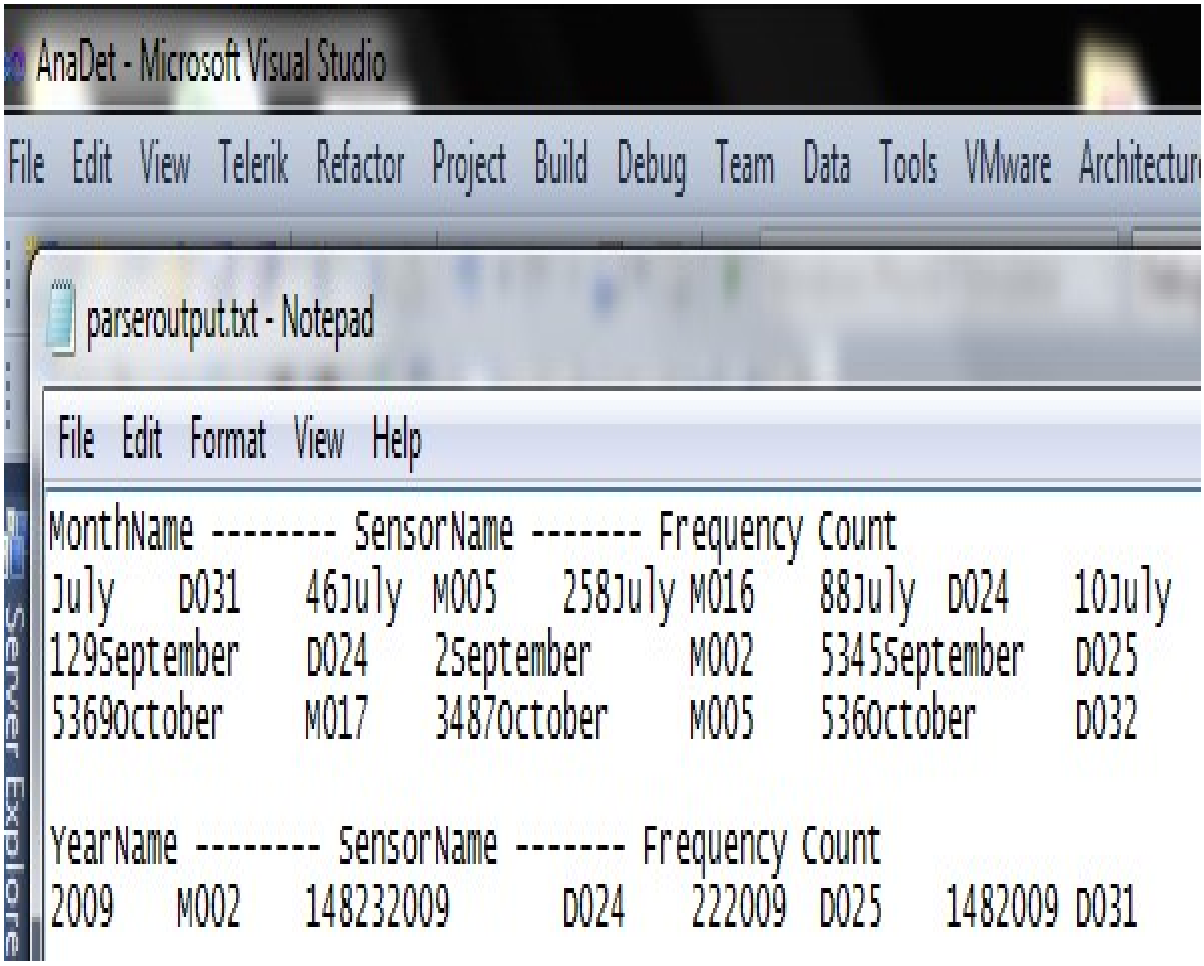


Figure 10 Activity Data Analyzer output results display

The Sensor Data from the smart environment is a bunch of sensors being on and off. This data needs to be annotated to get the mappings of sensors with the activity. Later this data needs to be analyzed for either patterns of trying to make sense of this data for automation to make lives easier. In the figure 10 the activity data is display by the month sensor name and frequency count.

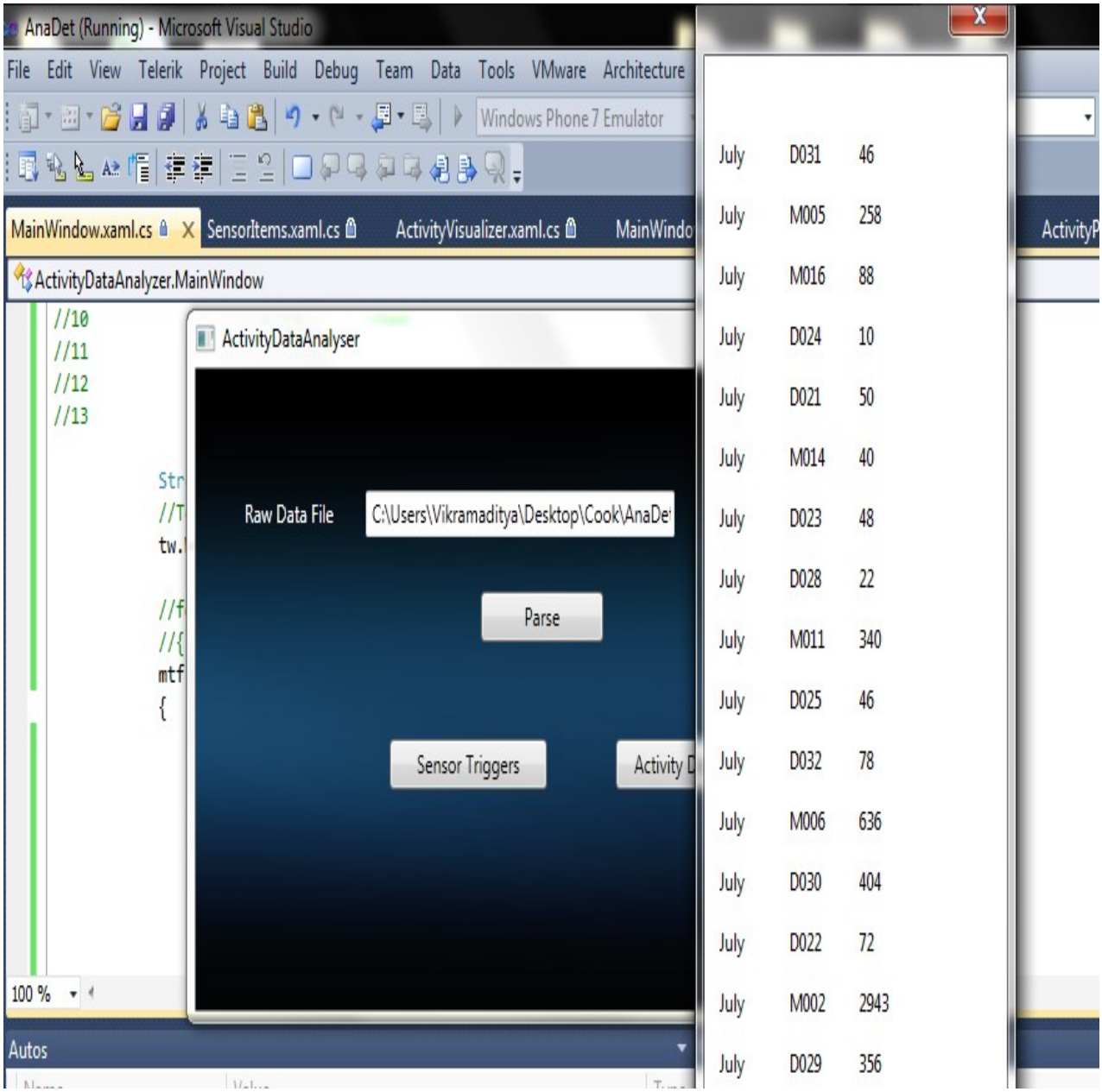


Figure 11. Display of Sensor Triggers.

The other view of data is the sensor triggers. Here we see the sensors triggered over a period of a month or month wise. It gives us an overview of the activity in the smart environment.

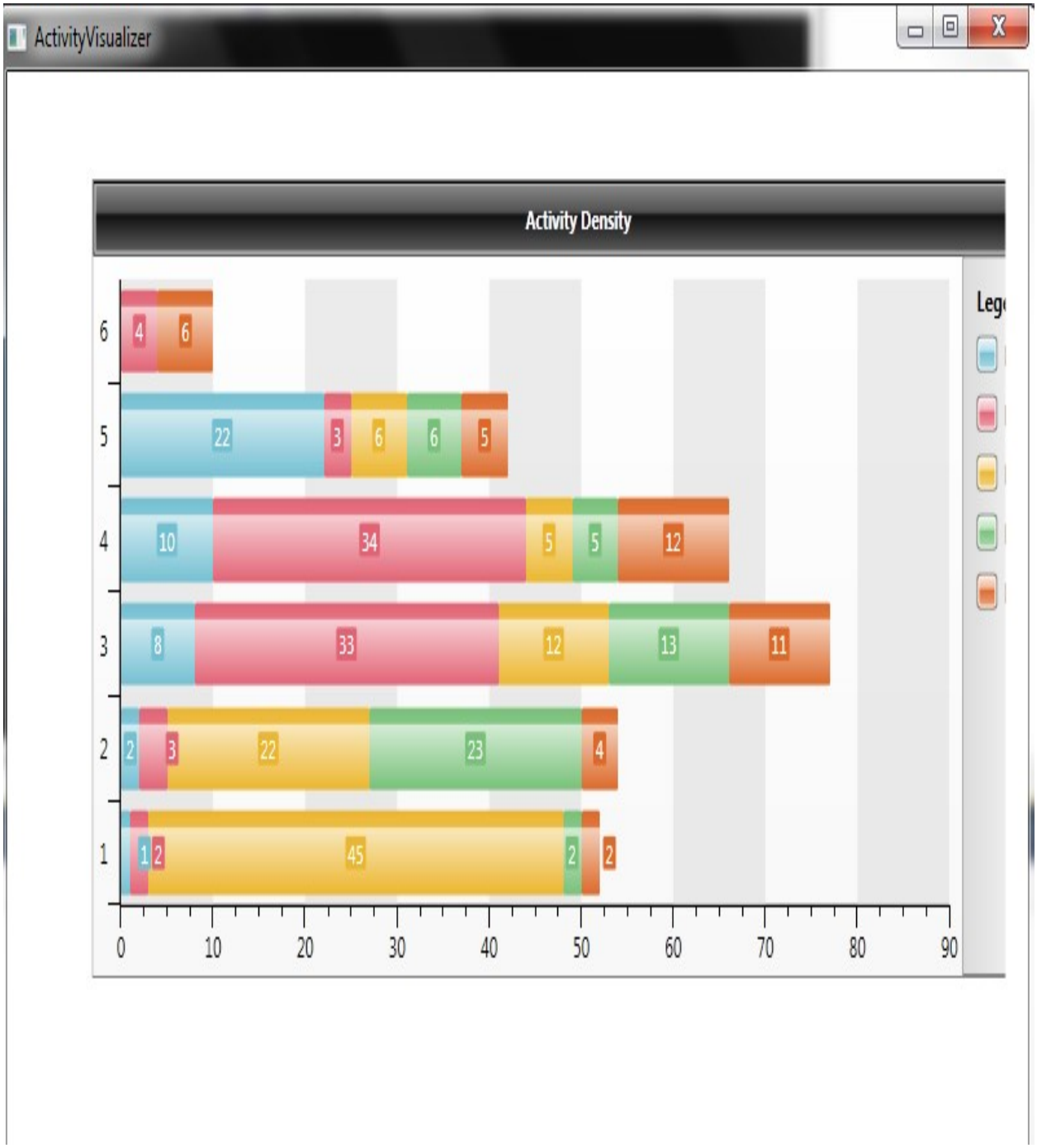


Figure 12: Activity Density

The activity density gives us overview of the activities in the house. You can see the bar chart and see the activity and its counts. Fox example you can see that the count and colors of different activities on a day in the figure 12 above.

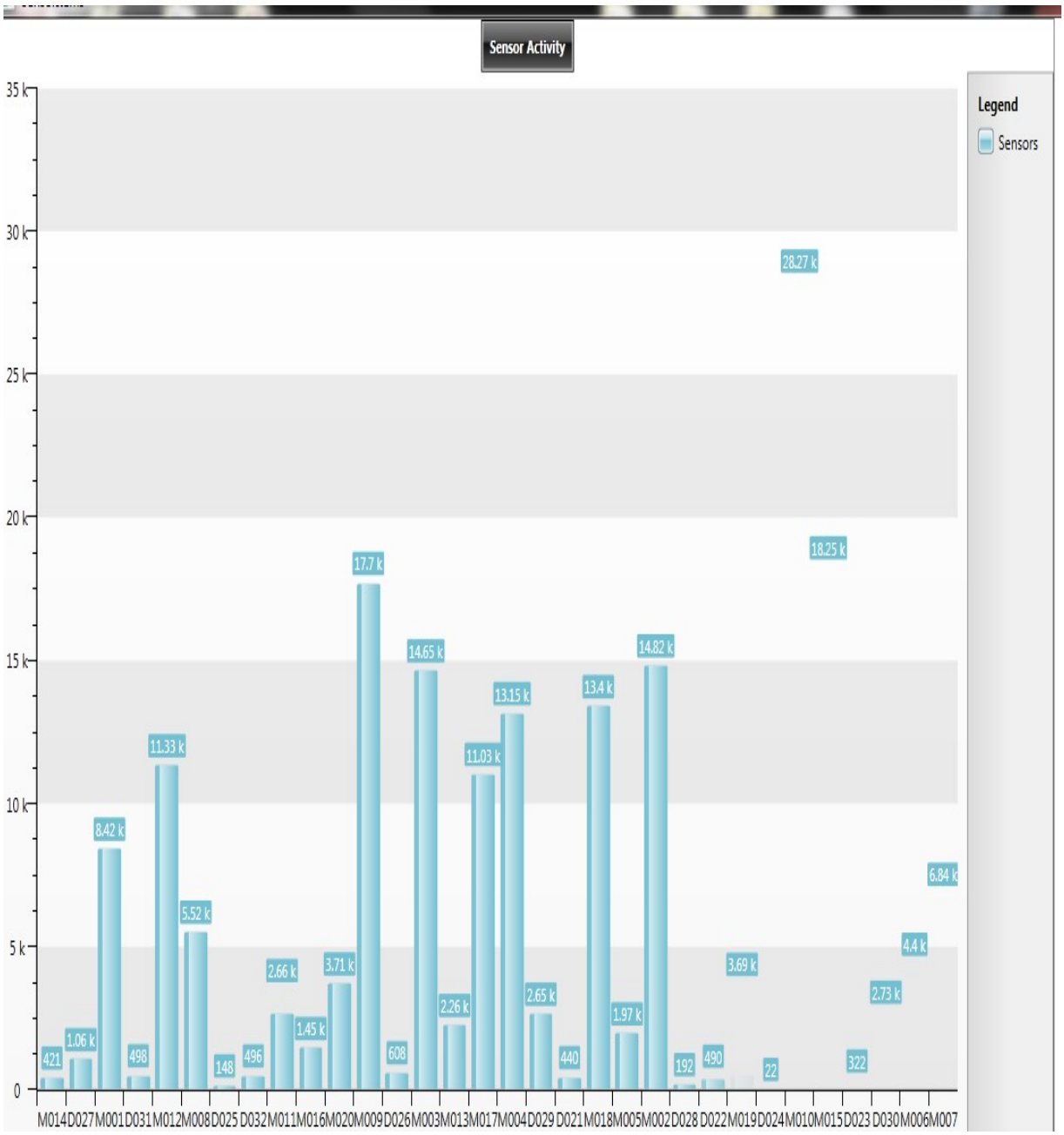


Figure: 13 Sensor Activity

This tool displays the sensor activities by the sensors for a fixed duration like a month or a day or a year based on settings.

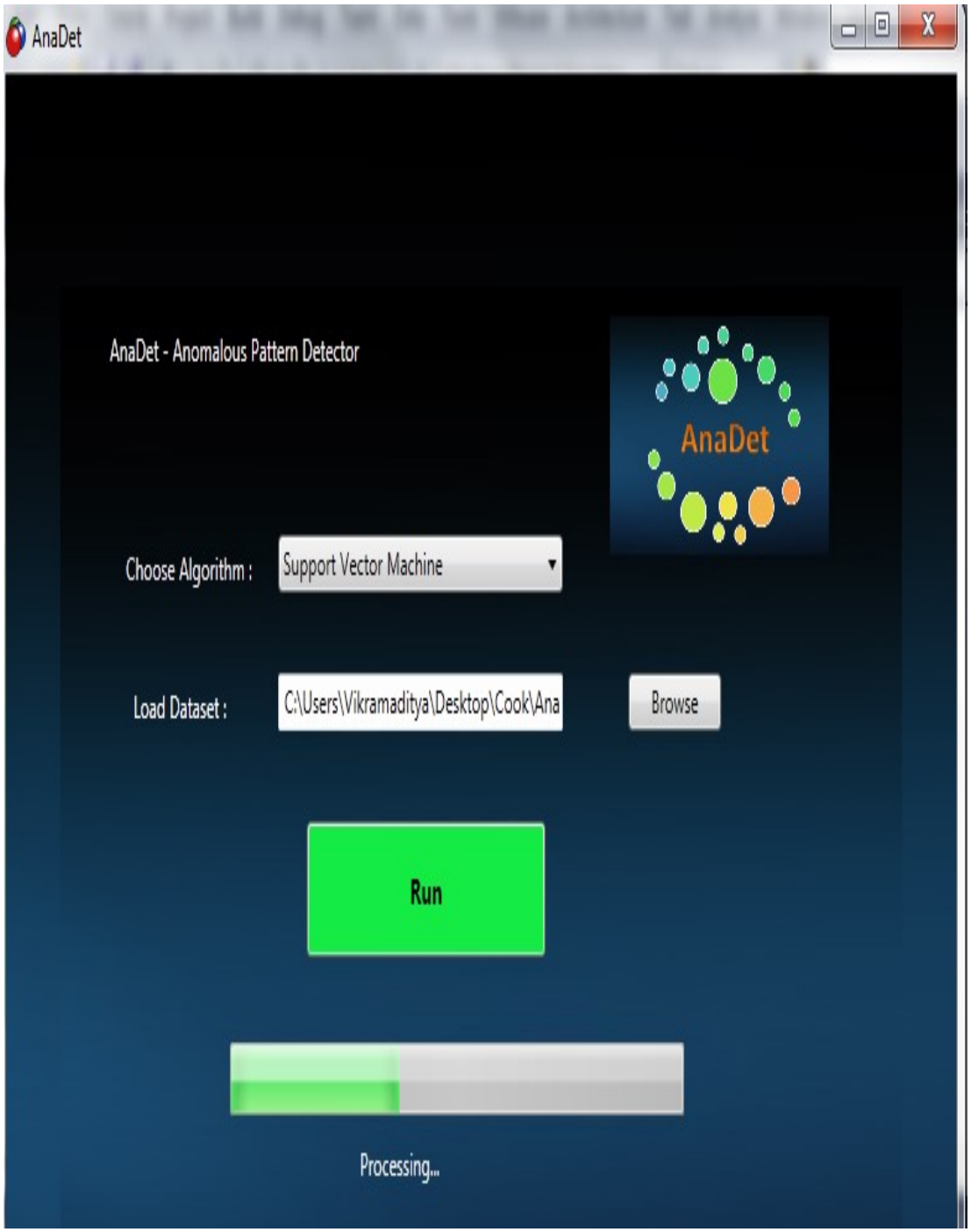


Figure 14 Support vector machine implementation

The support vector machine is implemented here. We see that it has both one class and multi-one class implementations.

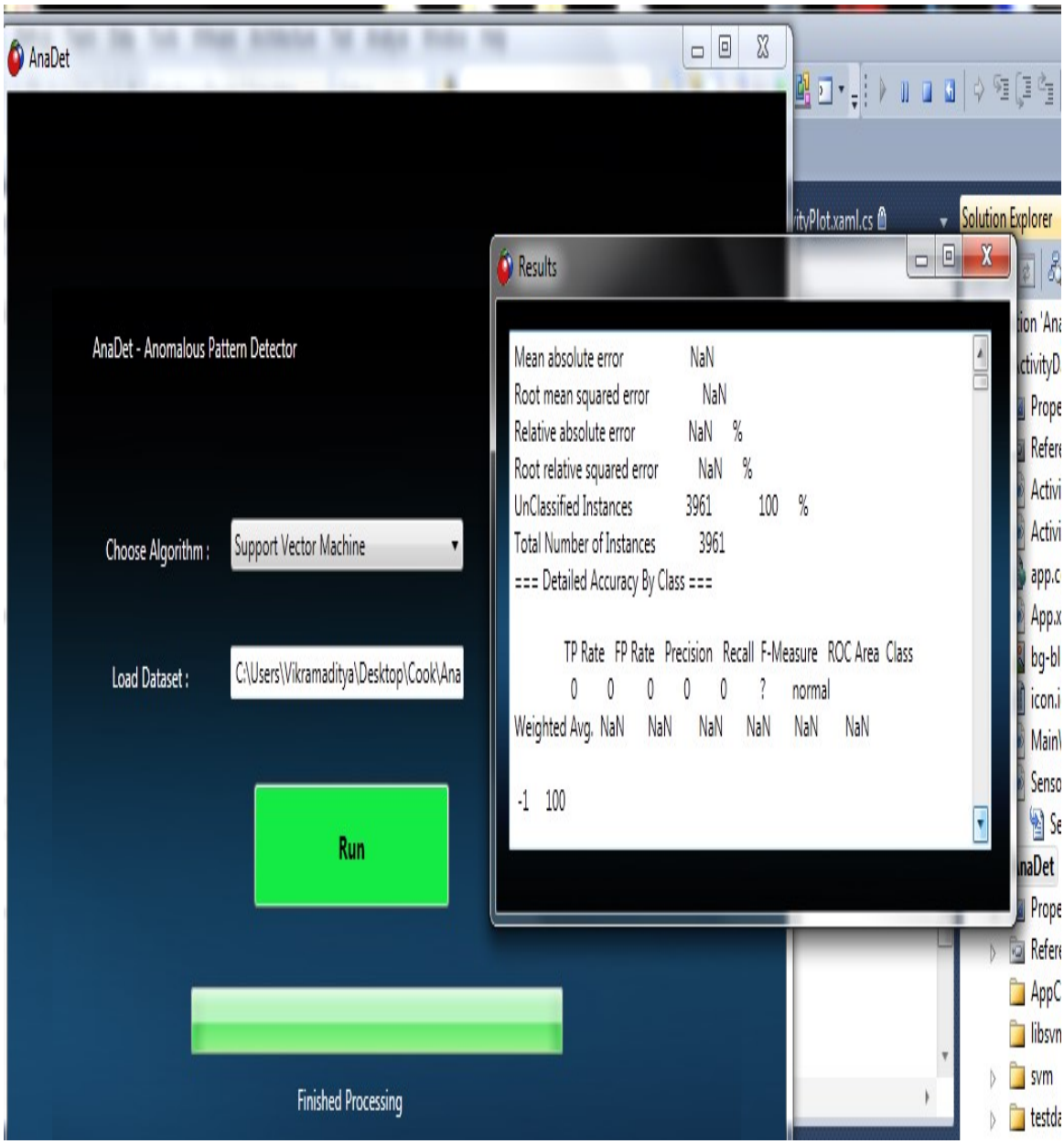


Figure 15 AnaDet Experimentation Results Screen

The support vector machine technique run results are displayed here. In this we see a flash screen and in that we see all the results and the value of the errors.

The earlier experiments consisted of running One Class Support Vector Machine (OCSVM) and Multi-One Class Support Vector Machine (MOCSVM). We have also run other techniques and have reported the findings below.

Source code Sample: Python Script.

```
  anomaly_output=append(anomaly_output,1)
    else:
     anomaly_output=append(anomaly_output,0)
    anomaly_cols=anomaly_data.size
    """
  print("anomaly_data.size=",anomaly_data.size)
  print("anomaly_data.shape",anomaly_data.shape)
  print("anomaly_output.size=",anomaly_output.size)
  print("anomaly_output.shape",anomaly_output.shape)
  """

  xcols=range(0,anomaly_cols)
  for x in xcols:
   #print(x,anomaly_data[x],anomaly_output[x])
   if((anomaly_data[x]==0)and(anomaly_output[x]==1)):
   fp=fp+1
 if((anomaly_data[x]==1)and(anomaly_output[x]==0)):
    fn=fn+1
if((anomaly_data[x]==0)and(anomaly_output[x]==0)):
```

```
    tp=tp+1

if((anomaly_data[x]==1)and(anomaly_output[x]==1)):

    tn=tn+1

  print("Type1 Error= ",fp/float(anomaly_cols))

  print("Type2 Error= ",fn/float(anomaly_cols))

  print("TN= ",tn/float(anomaly_cols))

  print("TP= ",tp/float(anomaly_cols))

  print(anomaly_output)

  return
fp/float(anomaly_cols),fn/float(anomaly_cols),anomaly_output,tn/float(anomaly_cols),tp/float(a
nomaly_cols)

probas_=testing(test,training(train,train_label),1,0)

rows,cols=probas_.shape

#fpr, tpr, thresholds = roc(label[half:], probas_[:,1])

xcols=range(1,cols)

# Logging to text file

log = open("log.txt", 'w')

log.write("MOC SVM Models\n")

log.write(str(training(train,train_label)));

log.write("\nProbability\n")

log.write(str(probas_));



log.write("\n:: Metrics ::\n")

"""

for x in xcols:
```

```
 fpr, tpr, thresholds = roc(test_label, probas_[:,x])
 roc_auc = auc(fpr, tpr)
 print "Area under the ROC curve : %f" % roc_auc

 print("fpr=",fpr)
 print("tpr=",tpr)
 print("thresholds=",thresholds)
 #MLUtility.plotROC(fpr,tpr)
"""
"""
log.write("Area under the ROC curve :",roc_auc);
log.write("fpr=",fpr);
log.write("tpr=",tpr);
log.write("thresholds=",thresholds);
"""
fp,fn,anomaly_out,tn,tp=anomaly(test_anomaly,probas_,0.9)
log.write("\nFinal Output\n")
log.write("\nType1 Error: ")
log.write(str(fp))
log.write("\nType2 Error: ")
log.write(str(fn))
log.write("\ntn : ")
log.write(str(tn))
log.write("\ntp : ")
```

```
log.write(str(tp))

log.write("\nAnomaly Output: ")

log.write(str(anomaly_out).replace('[',' ').replace(']', ' '))
log.close()
```

Exp 1: Running OCSVM with varying kernels & parameters

Table 4: Displays accuracy in % for the data sets.

Data\Kernel	Linear	Polynomial	Radial Basis Function	Sigmoid
Synthetic	49%	49%	52%	95%
B 1	49%	48%	50%	99%
B 2	50%	49%	41%	99.37%
B 3	49.62%	51.48%	50.52%	99%

Exp 2: OCSVM on frequent pattern data with Varying Kernel Parameters: nu

Table 5: Displays accuracy in % for the datasets. (Accuracy is number of correctly classified instances)

	Nu = 0.2			Nu = 0.5			Nu= 0.9		
Data/ Kernel	B	B	B	B	B	B	B	B	B
Linear	78	80	78	52	51	51	11	10	10
Polynomial	78	79	78	52	51	49	11	10	11
Radial Basis Function	2	6	3	0	2	5	0	2	0
Sigmoid	34	0	0	45	41	13	9	9	0

Exp 3: MOCSVM on frequent pattern data

Table 6: Displays accuracy in % for the datasets. (Accuracy is number of correctly classified instances)

	Synthetic	B 1	B 2	B 3
Training & Test split (80%)	69	70	72	70

Exp 4: Varying learning methods for anomaly detection

Table 7: Displays accuracy in % for the datasets. (Accuracy is number of correctly classified instances)

Classifier	Synthetic	B1	B2	B3
OCSVM	50	49	50	49.62
MOCSVM	69	70	72	70
Local Outlier Factor(LOF)	90	92	90	91
Local Density Based Clustering Outlier Factor	91	91	90	91

Experiment 3: Graph based approaches

The graph based techniques for anomaly detection:

- GBAD
- Nearest Neighborhood Score Approach

GBAD

We did identify that GBAD is used for anomaly detection in graphs by identifying substructures which are anomalous in nature. GBAD using MDL for finding normal patterns and find deviations and report them as anomalies based on the threshold passed.

For initial data processing, we built a sensor data to graph format conversion tool, which converted sensor datasets into subdue ready format which can be used by the gbad tool as input.

We converted the data based on activity annotation into sub graphs and feed in into GBAD and observed the anomalies reported by the tool.

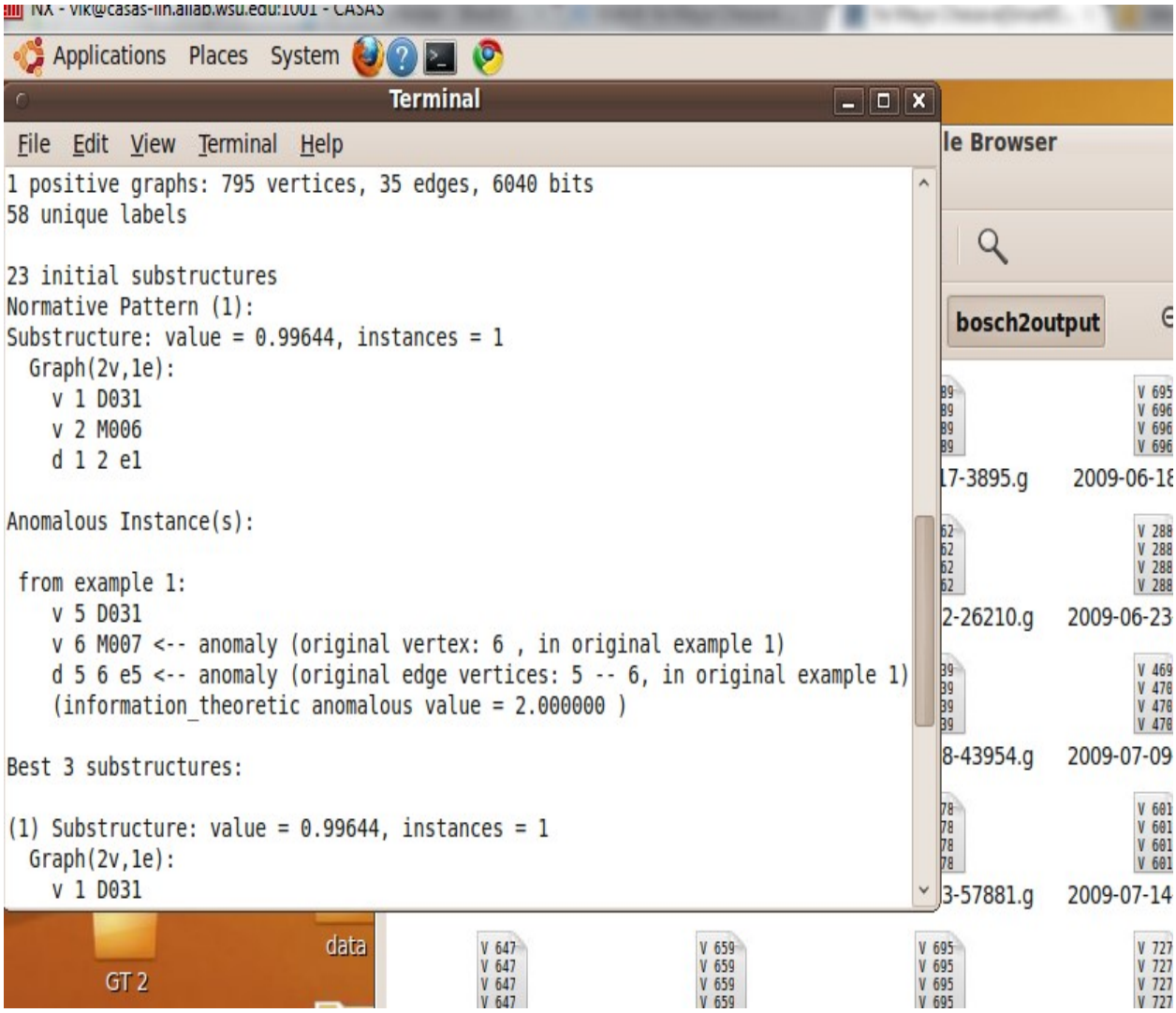


Figure 16: This figure illustrates a run of GBAD.

Algorithm

- Use MDL based approach for identifying anomalies.
- Break datasets into graph format based on activities and no annotations. + Use Anomaly tagged datasets as well.
- Observe report anomalies.

Nearest Neighborhood Score Approach

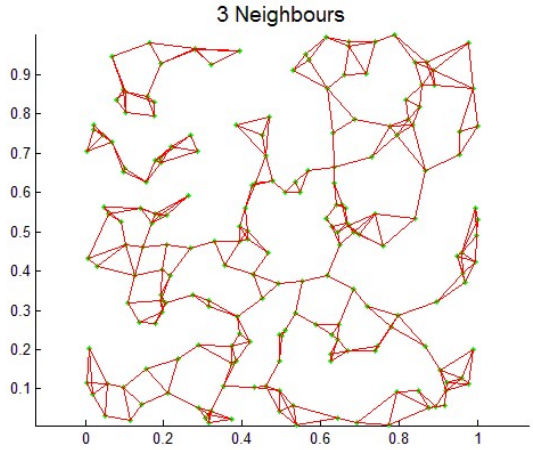


Figure 17 Nearest Neighbor Graphs Visualization

- Find all nearest neighbors in NO annotation data to fixed annotations graphs.
- Construct nxn Matrix with similarity score against each activity as column and no annotation as row.
- Get a combined score.
- Define a threshold and report anomalous observations

Algorithm

1) Break data into Annotated data and No annotation data.

2) Run with a distance metric on NO annotation data to form nearest neighborhood graphs.

3) Define an anomaly threshold

4) Annotation data is formed into activity graphs.

5) Now use a similarity or score metric to formulate each nearest neighborhood graph is from the annotated graphs.

6) If the score for a certain NO annotated graph is less than threshold for all Annotated graphs, we report it as anomaly.

CONCLUSION

Current research is oriented on activity classification and techniques that determine whether observed behavior falls into predefined sets of activities. Other research is focused on finding anomalies in predefined scenarios. However, both of them are limited to a small set of specific activities, which do not correspond to real behavior of the user at home. Techniques like automatic inactivity detection algorithm lack some additional information about the context of the user. For example when the user goes to sleep late, it is assumed that he will sleep longer, though this information is not incorporated in this technique. Our aim is to create a dynamical model that will adapt to changing conditions in the user behavior by the knowledge of some dependencies between the activities. For this purpose a Bayesian networks seems to be a good solution. In comparison with aforementioned models, we would like to focus not only on one specific abnormal activity, but to create a model from which more abnormalities may be acquired.

In today's world living smarter is more meaningful than ever. Long lasting and sustainable living is possible thanks to technology in everyday life. A robust anomaly detection framework is a niche area, and the resulting tools from this research area may be used to enhance the overall experience in a smart home setting by maximizing user adaptation, identifying issues in lifestyle, raising alerts, enhancing reminder system, and assist prompting systems. The approach in this paper is an initial step towards anomaly detection in smart home data which looks promising. Some future steps would include increasing the dimensionality of the SVM and to evaluate the use of the multiple one class support vector machines approach where we build an one class support vector machine for each annotation and see if an event is anomalous or not.

Anomaly detection adds value to smart home systems and has immense potential for a smarter living framework.

REFERENCES

1. Diane J Cook, Lawrence Holder. Mining Graph Data.

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.

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.

67. D. Cook and S. Das. 2004. Smart Environments: Technology, Protocols and Applications. Wiley Series on Parallel and Distributed Computing. Wiley-Interscience.

68. S. Deleawe, J. Kusznir, B. Lamb, and D. Cook. 2010. Predicting air quality in smart environments. Journal of Ambient Intelligence and Smart Environments, 2(2):145-154.

69. D. Cook, M. Youngblood, I. Heierman, E.O., K. Gopalratnam, S. Rao, A. Litvin, and F. Khawaja. 2003. Mavhome: an agent-based smart home. In Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, pages 521-524.

70. S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen. 2005. The gator tech smart house: A programmable pervasive space. Computer, 38(3):50-60.

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

72. G. Abowd and E. Mynatt. 2004. Smart Environments: Technology, Protocols, and Applications, chapter Designing for the human experience in smart environments. pages 153-174. Wiley.

73. Vikramaditya R. Jakkula, Diane J. Cook, and Aaron S. Crandall. 2007. Temporal pattern discovery for anomaly detection in smart homes. Proceedings of the the 3rd IET International Conference on Intelligent Environments (IE 07), Germany.

74. Hui-Huang Hsu, and Chien-Chen Chen. 2010. RFID-based human behavior modeling and anomaly detection for elderly care. Mobile Information Systems. Volume 6 Issue 4 341-354.

75. Vikramaditya Jakkula and Diane J. Cook. 2008. Anomaly Detection Using Temporal Data Mining in a Smart Home Environment. Methods of Information in Medicine,Smart Homes and Ambient Assisted Living special issue.

76. Ahmad Lotfi, Caroline Langensiepen, Sawsan M. Mahmoud, and, M. J. Akhlaghinia. 2011. Smart homes for the elderly dementia sufferers: Identification and prediction of abnormal behavior. Journal of Ambient Intelligence and Humanized Computing. Springer-Verlag.

77. Goldgof, D.B., Sapper, D., Candamo, J., and Shreve, M. 2009. Evaluation of Smart Video for Transit Event Detection/ Report No. 2117-7807-00 prepared by National Center for Transit Research, for Florida Department of Transportation and Research and Innovative Technology Administration.

78. An C. Tran, Stephen Marsland, Jens Dietrich, Hans W. Guesgen, and Paul Lyons. 2010. Use cases for abnormal behaviour detection in smart homes. In Proceedings of the Aging friendly technology for health and independence, and 8th international conference on Smart homes and health telematics (ICOST'10), Yeunsook Lee, Z. Zenn Bien, Mounir Mokhtari, Jeong Tai Kim, Mignon Park, Jongbae Kim, Heyoung Lee, and Ismail Khalil (Eds.). Springer-Verlag, Berlin, Heidelberg, 144-151.

79. B. Schölkopf, J. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. 2001. Estimating the support of a high-dimensional distribution. Neural Computation, 13, 1443-1471.

80. B. Schölkopf, A. Smola, R. Williamson, and P. L. Bartlett. 2000. New support vector algorithms. Neural Computation, 12, 1207-1245.

81. Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. ACM Comput. Surv. 41, 3, Article 15 (July 2009), 58 pages. DOI=10.1145/1541880.1541882 http://doi.acm.org/10.1145/1541880.1541882

82. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten . 2009. The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.