This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2668061, IEEE Internet of Things Journal

Survey on Prediction Algorithms in Smart Homes

Shaoen Wu, *Senior Member, IEEE*, Jacob B. Rendall, Matthew J. Smith, Shangyu Zhu, Junhong Xu, Honggang Wang, *Senior Member, IEEE*, Qing Yang, *Member, IEEE*, Pinle Qin

***Abstract*—The world has entered into a "smart" era. One area becoming smart is the place where we live – homes. Smart homes are expected to be equipped with numerous sensors to continually monitor, sense and actuate the space. The data from these sensors can be used to provide various types of services by automating common tasks while causing minimal disruption to daily life. In order to provide these services, a system must have sufficient intelligence to predict future events based on its observations. This paper first examines the requirements for smart home predictions. It then comprehensively reviews prediction algorithms and variations that have been proposed and investigated in smart environments, such as smart homes. It is these prediction algorithms that provide the intelligence required by a smart home. Comparisons are also made upon these prediction algorithms on their features and models.**

I. Introduction

A prediction algorithm uses information from past experiences to anticipate future events. These algorithms can be used in situations where data from a specific sample space can be collected over a period. This makes prediction algorithms ideal for use within smart environments. The term smart environment can be used to refer to homes, office buildings, power grids, etc. In this paper, the home context should be assumed unless stated otherwise.

A smart home can be defined in many ways. One definition is dwelling that contains sensors and device controllers in order to enhance one or more aspects of inhabitants' lives [1]. However, this definition does not fully encompass what a smart home is or how it operates. Another potential way to describe a smart home is that it consists of three main components. One component is an internal network, which is a way of communication such as wires or cables. Another component is an intelligent control, or an interface that allows for management of the system. The final component is home automation, where any product contained in the home is connected in some way to a service or system from an outside source [2]. In this type of smart home, the main point of emphasis is the use of remote access to activate or control a part of the smart home. However, what about having a smart home that can act on its own? This is the direction that researchers have been leaning toward, so there must be some way to allow the smart home to think for itself in order to complete certain tasks.

When smart homes were first introduced, the idea was to have a system in place for a resident to interact with the system through an interface to change some aspect according to their own preference. Though this is still a key component for a smart home, progress continues to be made to make the system more automated. This has led to researchers trying to implement an artificial intelligence into the environment to make decisions based on previously gathered information or prior inputs from the resident [3]. With artificial intelligence in play, this should allow a smart home to act on its own. This all seems ideal, but there still needs to be a way for the smart home to make suitable decisions. To do so, prediction algorithms have been introduced in many smart home systems to analyze information and make a choice that is appropriate for situation at hand.

One example of a smart home that has been developed to act more or less independently is the Managing an Adaptive Versatile Home (MavHome) [1], [4]. The researchers at the University of Texas at Arlington created this smart home project to have the home act as a "rational agent", meaning that the MavHome wants to make the resident as comfortable as possible while operating at maximally efficiency. The MavHome architecture is similar to many other smart homes in that it employs four layers: *decision, information, communication*, and it physical layers. The decision layer uses knowledge given to it from the information layer to select what action needs to take place. The information layer gathers and stores all data pertinent to any decision making process. The communication layer enables the flow of information or request between any two parts of the system. Lastly, the physical layer is all the hardware that is included within the smart home environment. MavHome uses this structure, along with prediction algorithms, to enable the smart home to be a more automated space. Some other work on this interest can be found in [5]–[14].

This paper is a survey on several different prediction algorithms that have been or are being used at smart homes. It will discuss the different areas within the home environment that use prediction algorithms. It will also compare the merits and

Shaoen Wu is adjunct with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044 China, and affiliated with the Department of Computer Science, Ball State University, Muncie, IN 47306 USA. Email: swu@bsu.edu.

Jacob B. Rendall, Matthew J. Smith, Shangyu Zhu, and Junhong Xu are students with the Department of Computer Science, Ball State University, Muncie, IN 47306 USA.

Honggang Wang is with Department of Electrical and Computer Engineering, University of Massachusetts Dartmouth, Dartmouth, MA 02747 USA. Email: hwang1@umassd.edu.

Qing Yang is with Department of Computer Science, Montana State University, Bozeman, MT 59717 USA. Email:qing.yang@montana.edu.

Pinle Qin is with School of Computer Science and Control Engineering, North University of China, Taiyuan Shanxi 030051, China. Email:qpl@nuc.edu.cn.

flaws of the algorithms presented and possible adaptations to enhance the overall quality of the prediction algorithms used in smart homes.

The rest of the paper is organized as follows. We first introduce the background of prediction algorithms, including system models and data in Section II. Then, section III extensively reviews the literature prediction algorithms, compares them upon their features and models. Next, Section IV discusses the enhancements and variants of the prediction algirhtms. Finally, the paper is concluded by section V.

II. Background

A. System Models

In order to better understand how prediction algorithms function, one must first understand the context in which they are used. The idea of predicting how a person will interact with an environment is a hard concept to grasp directly. This is why certain statistical models are typically used within prediction algorithms to better show how different tasks may predict future events within a system. A major benefit to modeling the system is that based any all data collected by sensors, it is not likely that this will provide the exhaustive set of activity sequences that can possibly occur within the smart home [15]. Similarly, probabilistic models tend to be even better than pure statistical models since they allow for some randomness, time complexity and dependency on other variables, and are therefore more fitting to how human behavior within a home can and should be modeled [16]. The main purpose of using models is to more accurately predict possible outcomes based on previous events occurring, and the most commonly referred to models used by prediction algorithms within smart homes are the Markov model and Bayesian networks.

The Markov model has been a well-established model for many years. The model says that at any given point in time of a system, the state at which the system is just depends upon the direct previous state and independent of all other states beforehand [17]. This has led many to describe a Markov model as a way to determine what is the ”most probable event sequence” for any given system [18]. This basically means the state of the system is recorded and then a sequence of states is developed over a certain time interval. From this, a probability can be estimated of what is most likely to occur next. This type of general Markov model leads to many different submodels that are also useful in prediction analysis. The most referenced Markov model in smart home prediction algorithms is the hidden Markov model (HMM) [19]. This specialized model is used to model human behavior and maintains the Markov property. However, HMM works on observations where the states of the system is unknown or hidden [17]. The idea behind this is that even if a system is missing a piece of data at any given time, it can use previous knowledge to make an educated decision on how to continue processing. HMM could also be considered to consist of three parameters: the initial probability of the beginning state, the transition matrix that shows the probability to change between states, and the observational model which gives the relation between two states [20]. A variation of HMM is known as the Task-based Markov model (TMM). In TMM, actions or observations form a cluster of data that can be treated as hidden [21]. So, the prediction analysis is still done in a similar fashion in TMM as in HMM, but with a different way of gathering initial states.

Similar to Markov models, Bayesian networks (BN) are also used to represent the probability of an event occurring based on previous observations. In fact, HMM is a model that is contained within the dynamic BN class, where the BN contains a time constraint [17]. Therefore, there must be some way to differentiate the BN structure from the original Markov model. Note that a BN is defined as a graphical model that demonstrates the joint distribution for any given set of random variables. So in any BN, the system can be expressed as a directed acyclic graph to demonstrate how different processes are connected and where dependencies exist [22]. It can be seen here how, if certain data has been collected to describe a sequence of events, a BN can be used (similar to Markov models) to find out what is the next likely event that will occur.

Though both HMM and BN are useful in modeling a system with missing information, there are potential problems in either model that can cause an error in the prediction of future events. For example, in HMM, there is no hierarchical structure and has trouble to process through a large amount of data [16]. This just means that HMM cannot determine what process or state is most important or what process must come first without being given the information explicitly. As noted before, a BN is a more general way of predicting human behavior than a Markov model, but a BN still has its own pitfalls. A major dilemma in using a BN is that it is rigid and has trouble adapting the “exact probabilistic inference” [16].. This refers to the inability of BN to determine what type of distribution is being used in the system. So when a prediction algorithm uses either of these types of models, there is a potential for error, though the error should be minimal as long as the model meets its required specifications.

B. System Data

1) Activities of Daily Living (ADLs): Prediction algorithms work on historical data to calculate potential future events. Each user’s actions within a smart home environment provide the data for the prediction algorithm to operate to its full capability. In general, these actions conducted by a user fall into the category known as Activities of Daily Living (ADLs).

An ADL can be one of many different activities, though gathering the information from these actions is typically performed by using a sensor network. The sensors that are used can be either wearable that go directly on a person, *a.k.a* wireless body sensor area network (WBSAN), or non-wearable that remain stationary in the environment. These sensors can collect the information that could include a user’s location, movements, or activities [23]. The use of ADLs at a smart home has its own challenges. One potential flaw is that though ADLs represent a sequence of events, some activities can occur in different orders or together with a set of other activities [24]. This means that any ADL, though beneficial to use in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2668061, IEEE Internet of Things Journal

predicting inhabitant activities, could lead to complexity since they are not necessarily identical for the same activity. Another possible problem is the issue of inter-subject variability, which refers to the situation that both inhabitant and non-inhabitant activities can happen and affect how the ADL is recorded, resulting in a potential error [25].

When a sensor gathers information on an ADL, not only does it get data related to the activity but will also record the timestamps of events occurring within the activity. So, although any given ADL occurs as a sequence of events, the use of the duration of each step along with the sequence will allow for better activity prediction [26]. Since there are times when sensors can malfunction or inhabitants can adjust their routine, the use of duration can help better compare an activity history. All of the ADLs can then be used by a prediction algorithm to allow the smart home to both help the user and make the house more comfortable. This also can help reduce the cost of certain activities by being better prepared. One example of this would be to reduce energy consumption. An experiment done by Zhuang *et al* demonstrates how the use of ADLs can benefit the smart home in its energy use [27].

Overall, an ADL is a resource that, when used, allows prediction algorithms to be extremely accurate in foreseeing when and where future activities could occur within a smart environment.

2) Location: The ability to predict the current location or the next location of inhabitants in a smart environment is important to the functionality of the system as a whole. It is generally not desirable to manipulate a portion of the environment based on a prediction if the inhabitants are not currently interested in that part of the environment.

One example of location prediction is performed by Petzoid et al. in the context of an office building [28]. Two major applications of next location prediction within an office are proposed. First, a device could be placed outside each office in the building to display the predicted location of a person when he is away from his office. This would enable any visitors to decide if they should go to the predicted location or return at a later time. The second application is call forwarding. It could prevent a missed call by forwarding the call to the predicted location when a person is away from his or her desk. Petzoid et al. have investigated Neural networks, Bayesian networks, Markov, and state predictors for creating a system with the described functionality. In addition to location, a prediction of length of stay is necessary for these applications. Petzoid et al. claim that this could be easily predicted with the same techniques that they have already investigated. In this scenario, it is also important to understand when no prediction is better than an incorrect prediction. Too many incorrect predictions could annoy users no longer interested in the predictions [28].

Weiser investigates the location prediction in a smart home environment. He envisions a smart environment with a high concentration of computing and communication capabilities and seamlessly integrated with its human users [29]. In order to provide information to the user, the system must be aware of the user's location. The information needed by the user is strongly related to the current or future location of the user [30]. Also, information must be communicated to the user by a device that is close enough for the user to notice it.

Roy et al. analyze daily routines of inhabitants to reveal that patterns exist in daily routines. They also realize that routines likely change over time; however, this change can be classified as infrequent and random. Additionally, Roy et al. note that a smart home usually contains multiple paths between two locations, but they observe that inhabitants usually exhibit a most likely path. With these observations, they assume inhabitant location to be a piece-wise stationary, ergodic, stochastic process. Based on this assumption, the previously discussed LeZi update algorithm is reasonable to be used for a prediction algorithm for this domain [30].

III. Algorithms

The architecture of smart homes is often represented as a layered model. A four-layered model is proposed by [31]. The top layer of this model is the "Decision Layer", which bears the responsibility of determining actions for the smart home to perform by using the data from other layers. The other layers generate data for the decision layer through communication with physical hardware and storing the information obtained.

In order to select actions, the decision layer implements one or more prediction algorithms, which will infer inhabitant actions to occur next with a large probability. Using the result of the prediction algorithms, the decision layer can choose to take one or more actions. One possible choice is to automate the predicted action. Another possibility is to deliver notifications or reminders if the predicted action is not performed within a specified time threshold. This section reviews some well-know prediction algorithms used in the literature of smart environments.

A. Active LeZi

Although Active LeZi has been a widely accepted compression algorithm, it is used in prediction the context of smart homes. One application of Active LeZi is in the MavHome project at the University of Texas at Arlington [32].

Active LeZi is an improvement on the LZ78 algorithm [33], [34] as proposed by Lempel and Ziv. By parsing an input string into one symbol at one time, LZ78 calculates the probabilities for the next symbol at each context in the data. LZ78 will generate probabilities for the contexts that it sees from the beginning of the data through the current symbol. The algorithm parses the input string into substrings "$w_1, w_2, ...w_n$" such that for any j¿0, the prefix (consists of all characters of w_j except the last character) of w_j is equal to some w_i. Because of this prefix property, the substrings can be stored in a trie structure. Since LZ78 was designed as a compression algorithm, it has both an encoder and a decoder. For the purpose of prediction, only the encoder is required because there is no need to produce the original data string from the trie structure which approximates a Markov model. An example of the results of LZ78 parsing can be found in [35].

LZ78 cannot be used in practice for two main reasons. First, data that crosses phrase boundaries is lost. This limits the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2668061, IEEE Internet of Things Journal

usefulness of LZ78 for prediction because potential patterns often occur across phrase boundaries. Secondly, previous work has shown that LZ78 does converge to optimal predictability, however, the convergence does not occur quickly enough. LeZi Update [36] was proposed, but did not completely solve the issues presented by LZ78. It marginally improves the slow convergence problem, but it does not address the loss of information across phrase boundaries.

In order to overcome these limitations, Gopalratnam and Cook introduce the Active LeZi to be used in smart environments. Active LeZi improves on LZ78 and LeZi Update by including a window that varies in length as the algorithm runs. The window holds previously observed symbols. The length of the window is determined by the longest phrase observed in traditional LZ78 parsing. The window makes it possible to gather information for all possible context and approximate an order-k Markov model where k is the length of the longest LZ78 string observed so far. As discovered, the Active LeZi trie contains more nodes than the LZ78 trie and thus produces a more complete Markov model [35]. The amount of information stored will grow with the length of the maximum LZ78 string, saying that algorithm performance will improve as the experience grows. Active LeZi converges faster than LZ78 because it gathers more data. The trie output of Active LeZi can be then used with Prediction by Partial Match (PPM) predictors. PPM predictors take the different order Markov models into consideration and weight them accordingly. They build a probability distribution for the next symbol at a given context. Then, they predict the symbol with the highest probability. Gopalratnam and Cook perform experiments on Active LeZi using two data sets. One data set was synthetically generated without noise. The other data set was typical scenarios from a smart home with noise. As the number of training instances increased, the performance on the noiseless data converged to 100% accuracy. For the noisy data, accuracy converged to 86%.

Fang and Ruan propose an enhancement to Active LeZi termed $Time-varyingLeZi$ (TALZ) [37]. The enhancement is based on the observation that human activity tends to occur on a periodic basis. For example, a person may wake up at 7:00 AM each morning, go to work at 8:00 AM and return for dinner at 6:00 PM. These three actions are repeated on a daily basis. Additionally, distinct behaviors could appear similar but occur at different times. For example, going to work and walking the dog could both involve turning off the lights and locking the door but occur at different times of the day. In order to improve Active LeZi, TALZ creates sub-trees from the null context based on the time of day. TALZ subdivides a day into 24 one-hour intervals, but this could be easily modified for any desired size of interval. Active LeZi parsing is used, but the output is placed into a sub-tree based on the hour of the day that the event occurred. The output is essentially that each sub-tree has a parent node corresponding to the hour of the day, and all of the hour nodes share the null context as a parent. Thus, the output is actually a single tree.

B. Flocking

Flocking is a term that is most commonly associated with animal species, more specifically birds. Flocking mainly consists of making groups of objects that share similar characteristics, and it has been used in the computer science to describe an algorithm that acts on a set of objects to group them according to their features [38]. To implement a flock, computer scientists have developed certain rules to govern what constitutes a flock or not. Reynolds came up with three specific rules that describe the behaviors of a flock, which he refers to as a boid [39]. His first rule is collision avoidance, or the chance to evade nearby obstacles or other members of a flock. The second rule is flock centering, or the need for all flock mates to remain close together. The final rule is the velocity of any flock member should be the same or close to the others around it. In other words, the whole flock must be moving toward the same direction [39]. In simpler terms, these rules are alignment, separation and cohesion. These rules have been used and adapted to software, especially in robotics and artificial intelligence for the ability to model not only the architecture of the environment, but the adaptability for a “flock” (or actions) to enhance the system.

Within a smart home, flocking algorithms are useful tools in analyzing the ADLs of an occupant, especially when said occupant may be suffering from a cognitive disability like dementia. Any given set of ADLs can act as a cluster, and then a flocking algorithm can be used to parse through copious amounts of information quickly and effectively. It should be noted that there are some other clustering algorithms that can work, such as the K-mean. However, K-mean clustering is not ideal for the smart home since it requires human validation and a given initial number of clusters to evaluate. Therefore, K-mean is not effective when trying to allow a smart home to run independently of human involvement with the system. This is partly what led to the idea of using a flocking algorithm as a clustering approach to the smart home since it does not require any initial number of clusters and will partition objects itself based upon previous knowledge gathered. Also, a flocking algorithm can adapt easily to the environment by either adding new cluster points or combining new data with its appropriate partition [40].

In a study conducted by Lapalu $et\ al$, they attempt to validate the use of a flocking algorithm within the smart home environment with some extra rules [40]. As stated earlier, the basis for any flocking algorithm is the set of three rules established by Reynolds. After following these rules, they find that the attempt to cluster objects could result in just one single cluster. So, they adapt the flocking algorithm by adding two new rules: similarity and dissimilarity. Similarity will bring similar objects together. Likewise, dissimilarity will attempt to keep all dissimilar objects away from one another. In following all five of these rules, the study demonstrates that the adapted flocking algorithm will develop a good clustering since it will break apart large clusters into smaller, more similar clusters [24]. This allows the algorithm to more easily identify ADLs that are alike and get relevant information that the system can use to adapt to make the home environment more suited to the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2668061, IEEE Internet of Things Journal

occupants need.

C. SPEED

In order to predict how inhabitants interact with a smart home, the algorithm called Sequence Prediction via Enhanced Episode Discovery (SPEED) was used. The main purpose of the SPEED algorithm is to learn from the periodic tendencies of the user of a smart home and attempt to make appropriate decisions based upon the data it has gathered. SPEED develops a finite-order Markov model to describe the usr activities and exploits the PPM algorithm to make predictions [41]. Namely, SPEED takes a sequence of events or occurrences from an inhabitant's interaction with the smart home to predict future events within the environment.

The SPEED algorithm employs decision trees to form a data structure where it is able to view previously learned information in order to make an appropriate decision. At certain points throughout the cycle, SPEED will discard any immediately unnecessary information in order to make a decision tree that contains at most k levels, which in turn represents a k-th order Markov model [41]. It should be noted that in using PPM to help make decisions, a weighted branch system is put in place which can even predict event sequences which have never been present in the system [42]. The SPEED algorithm is continuously calculating the weighted probability distribution after each successive decrease in the overall length of any given sequence in the decision tree [41].. This is fairly exhaustive and is expected to make a more accurate prediction of future events.

Now, although the SPEED algorithm is able to form sequences of events to help predict what is to occur, one problem is that it does not take into account the time of events. So, further study was performed to include time into SPEED, and the modified SPEED (m-SPEED) algorithm was therefore developed. The motivation was to make the algorithm more accurate, since, in many home environments, most tasks or jobs are done in the same time period. Also, m-SPEED tried to get rid of all false episodes that showed up in the original SPEED algorithm [42].

D. Nash H-Learning

Many learning algorithms can be useful in addressing how a smart home will learn and adapt to the changes that occur within its environment. If the system is able to learn habits of an occupant more effectively, then the more productive a smart home can function. One such learning algorithm that has been looked at as a potential solution to predict certain behaviors within a smart home is a method known as Nash H-learning. In order to understand how the Nash H-learning algorithm or framework functions, a small amount of background information must be given first.

Nash H-learning was developed from an algorithm called Q-learning, which uses a Markov process in order to come to a decision that is best utilized within a stationary environment. Although the Markov process is well-established as being useful within a smart home context, Q-learning itself has its flaws. The most important downfall of Q-learning is that it does not function properly on multi-agent systems since the environment is constantly changing due to outside objects as opposed to an initial stochastic process [43]. This leads to the idea of Nash equilibrium, which is viewed within a smart home as a balance between all preferences from users of the system, where no person leaves [44]. Expanding upon this, the Nash Q-learning takes the basic Q-learning method into a multi-agent system with the assumption that all choices lead to a Nash equilibrium [43]. It is from this Nash Q-learning idea that the Nash H-learning technique was developed.

The Nash H-learning framework (H stands for entropy) is mainly used to predict the location of multiple occupants of a smart home. Note that entropy, in this context, is referring to location uncertainty of any number of residents within the smart home. To predict the location, it looks at the correlation between all people's movements. This leads to the Nash H-learning algorithm to attempt to conserve energy by continuing to use the data extracted from the environment to a lower entropy. The lower the entropy, the more accurate the Nash H-learning algorithm is in predicting the correct location. Overall, this algorithm attempts to provide maximum comfort for the smart home users while also keeping the energy consumption to a minimum [44]. So the Nash H-learning technique just expands on the previous known learning processes to predict locations of multiple users, which then helps the smart home to better manage its own resources.

E. Apriori Algorithm

The Apriori algorithm is a classic data mining algorithm for mining association rules. As described by Schweizer, it traditionally runs on a set of database transactions in order to detect recurring patterns. The output of the Apriori Algorithm are association rules of the form "$\{X\} \rightarrow \{Y\}$" where "X" and "Y" are subsets of the set of all items from the input transactions. A rule where "X" and "Y" are both the null set is not a valid output. The rule is interpreted as follows: the occurrence of the items in "X" implies the occurrence of the items in "Y" [45].

Running the traditional Apriori algorithm is a two-step process. First, the raw data is processed into subsets termed as itemsets and "frequent" itemsets are identified. Second, the collection of "frequent" itemsets is pruned. The output, which is a set of association rules, is created from the remaining "frequent" itemsets after the pruning is completed.

In the first step of the Apriori algorithm, itemsets are generated starting with a size of 1 item and increasing until no larger subsets can be found. Each itemset's support is calculated, and the itemset is considered frequent if the support meets a predetermined support threshold. An itemset's support is defined as the percentage of the total input transaction that contains the given itemset. In the second step of the algorithm, association rules are created using the frequent itemsets as determined in the first step of the algorithm. Rules must meet a predetermined confidence threshold in order to be included in the algorithm output. Confidence is defined as the percentage of transactions containing "X" that also contain "Y".

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2668061, IEEE Internet of Things Journal

TABLE I: Summary of Prediction Algorithms

Algorithms	Data Structure	Model	Data	Category
Active Lezi	trie	Markov	location and time	LeZi
SPEED	tree	Markov	event sequence	Episode Discovery
Flocking	cluster	rules	ADL	clustering
Apriori	matrix	rules	temporal relation	AI
H-learning	users	Markov	location	Q-learning

The Apriori algorithm was proposed before smart home environments were heavily researched. As a result, smart home researchers have determined that its traditional implementation is not suitable for smart homes and proposed improvements. One of these proposals is made by Zhang and Qi who observed that this algorithm takes too long to run in order to be practical for a smart environment. In order to overcome this problem, they propose optimizations. First, Zhang and Qi observe that the standard implementation of the Apriori algorithm requires multiple scans of the database in order to obtain the required information. To address this, they propose scanning the database once and storing the pertinent data in memory. Then the algorithm can run on data in memory instead of taking the extra time to access a database. A second improvement is proposed based on the idea that a frequent itemset cannot be obtained by adding an additional item to a non-frequent itemset. This reduces the number of iterations that the support must be calculated for an itemset. Zhang and Qi include experimental data that shows the execution time for their enhanced Apriori algorithm is only a fraction of the execution time of the standard Apriori algorithm. Their results also show that as the size of the problem grows, execution time grows less quickly for their enhanced algorithm than the standard algorithm [46].

Rashidi and Cook discover another problem with the Apriori algorithm within a smart environment. The algorithm was originally designed to discover frequent itemsets, but a smart home environment also requires the ability to discover periodic itemsets. In order to address this problem, Rashidi and Cook design a variant of the Apriori algorithm that is called Frequent and Periodic Activity Miner (FPAM). It defines one hour and one day granules of time in order to recognize periodic activities. After FPAM determines its candidate itemsets, it classifies them as periodic or frequent. It is possible for an itemset to be both periodic and frequent; however, FPAM will report it as frequent only [47].

F. Summary of Algorithms

All prediction algorithms that are used within a smart home environment have a common goal, which is to allow the system to make an educated prediction based on previous information in order to make the users life easier and more comfortable. Even though this is the main role for every prediction algorithm, they differ in their approaches to storing, interpreting and mining data for the system. The following sums up these five algorithms presented above in both similarity and difference in many aspects, which is highlighted in Table I.

Information Presentation: The most widely cited prediction algorithm used in a smart home is the Active LeZi. This algorithm, as previously stated, is an adaptation of the LZ78 compression algorithm. As it parses the data, the Active LeZi generates a trie structure to hold the information. In a similar fashion, the SPEED algorithm also uses a tree structure that contains the information to predict future events. The other algorithms, however, do not share this same structure. In the flocking algorithm, all the data is stored in individual clusters that share common characteristics. On the other hand, the Apriori algorithm operates on transactions stored within a database. An optimization of the Apriori algorithm loads the database transactions into a matrix structure in the memory to avoid the time required for disk access. As for the Nash H-learning algorithm, it is also unique in how it retains information. Instead of centralizing all the data into a single source, this algorithm keeps all the preferences and information on the individual users themselves before going through to find what is most likely to occur.

Model: One major commonality among any prediction algorithms is whether any of them relies on a Markov model or Bayesian network to function. Of the algorithms discussed in this paper, both the SPEED and Active LeZi algorithms are based on Markov models and use a PPM approach to generate predictions. However, they determine the size and order of these models in a slightly different manner. For Active LeZi, the order of the Markov model is determined by the length of the longest LZ78 string observed. However, for SPEED, the order is determined by the number of levels in the decision tree. Also, note that the Nash H-Learning relies on a Markov model since the Q-learning technique is adapted from a Markov model and Nash H-learning is itself an adaptation of Q-learning. The other two mentioned algorithms operate without a Markov model or Bayesian network and rather work on rules. In order to determine its results, the Apriori algorithm evaluates the candidate rules that it generates based two metrics, namely support and confidence which are defined above. The flocking algorithm uses three rules in order group similar objects together: alignment, separation, and cohesion.

Data Type: Just as the algorithms structures differ at times, so does the information that they are gathering. They all retrieve data in order to predict future events, but they most definitely do not all get the same data. For the Nash H-learning algorithm, it is mainly concerned with getting the location of all users, and then uses this to try and conserve energy. Going a slightly different route, the flocking algorithm instead tries to get all information pertaining to each users' daily activities (or ADLs). This can allow the smart environment to adapt to fit the users' needs. The SPEED algorithm instead takes data in terms of sequences of events. From this, it attempts to predict potential events that will occur in the future. However, the SPEED algorithm does not take time into consideration when doing this. Yet, note that the Apriori algorithm is often used in looking at temporal relations. Now, the Active LeZi algorithm does not focus on only one area to gather data from. Instead, it can be used to get information on location and time that certain activities occur. As can be observed, although there is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2017.2668061, IEEE Internet of Things Journal

some overlap, these five algorithms gather data that pertains to various aspects of the environment.

Genealogy: It seems common for a prediction algorithm to be a modification or improvement of an existing algorithm. For example, Active LeZi is an improvement to LeZi Update which is an improvement to LZ78. Extensions exist for Active LeZi and SPEED in order to take time into consideration. Along a similar line of thought, both the flocking and Nash H-learning algorithms are also adapted slightly from previously known works. Flocking is a clustering algorithm, though it does not require knowing the number of clusters up front, unlike other known clustering algorithms. Also, Nash H-learning, as pointed out earlier, is adapted from the Q-learning technique to work on multiagent systems or environments. The Apriori algorithm has shortcomings that prevent it from being used in its native form in a smart home environment; however, it becomes practical with certain optimizations.

IV. Algorithm Enhancements

Some ideas and concepts have been introduced, which however cannot stand alone as complete prediction algorithms. They are designed to be used in conjunction with a prediction algorithm. While these enhancements do not fall into the category of prediction algorithms, they aim to increase performance when paired with a prediction algorithm. This section reviews these enhancements.

A. Episode Discovery

Episode Discovery is a data-mining algorithm that searches for behavioral patterns in a string of data [48]. Heierman and Cook have proposed a framework for smart homes based on this algorithm [49]. The collected raw data does not indicate where periods of device interaction, or episodes, begin and end. The framework searches for the beginning and end of episodes while balancing episode length with frequency and regularity. This approach discovers significant episodes that occur on a recurring basis such as a daily wake-up routine or weekly lawn watering.

The episode discovery algorithm begins by partitioning the input stream of events into maximal episodes. An episode window is used to enforce the rule that all events in a maximal episode must occur within a specified time period. From the maximal episodes, itemsets are created. These itemsets include the maximal episodes themselves as well as subsets of the maximal episodes. Next, the itemset space is pruned. After pruning, the remaining itemsets are considered significant episode candidates. A significance value is computed for each candidate, and significant episodes are chosen using a greedy approach with a minimum compression threshold.

This method improves prediction algorithms by filtering the input stream before it is provided to the prediction algorithm as training data. Heierman and Cook identify two algorithms that expect to perform slowly in smart home environments due to the noisy training data: Incremental Probabilistic Action Modeling (IPAM) and Back-Propagation Neural Network (BPNN) [49]. Episode Discovery was evaluated in a case study upon these two prediction algorithms. IPAM improved from 41.0% accuracy to 73.6% accuracy with the addition of episode discovery in the case study. Similarly, the accuracy of BPNN improved from 63.6% to 85.6% [49] .

B. Temporal Relations

One potential flaw in smart home prediction is the modeling of an activity occurrence as a single point in time. In actuality, an activity spans an interval of time with a starting point and an ending point. For example, an activity may start when the living room light is turned on. As the activity progresses, the TV is powered on and later powered off. The living room light is powered off which ends the activity [50].

Jakkula and Cook propose a framework for smart homes based on temporal logic [51]. They base their framework on the thirteen temporal relations included in Allen's temporal logic: before, after, meets, meet-by, overlaps, overlapped-by, starts, started-by, finishes, finished-by, during, contains, and equals. This provides additional insight into the timing of events than a stream of symbols which represent individual events which can enhance prediction algorithms. With this information, an algorithm is able to create predictions with temporal constraints resulting in better detection of anomalies. For example, if event A usually overlaps event B, event B occurring after event A would be considered an anomaly. Also, temporal relations allow for the calculation of the probability of event X based on the occurrence of events temporally related to X [51]. In the knowledge discovery and pattern recognition of smart homes, this approach faces some challenges as noted by Jakkula and Cook [52]. One challenge is the ambiguity of Allen's relations that it is possible to have two events that can be correctly described by more than one temporal relation. Allen did not define a method of choosing a single "best" relation for a given set of events. In order to address the ambiguity problem, Jakkula and Cook define boundary conditions involving the start and end times of the related events for each of the thirteen temporal relation. Their proposed algorithm relies on these boundary definitions in order to identify the relationship of two events [52].

Another problem is about the highly detailed time data, which is common in smart environments. In order to solve this problem, Jakkula and Cook use an application of the Apriori algorithm to identify associations between events. The Apriori algorithm mines the data for sets of events that happen together frequently. It returns sets of events that meet a minimum support threshold. In many cases, the returned frequent sets of events will have multiple temporal relations. In this case, the relation that occurs at the highest number of times is chosen.

In another work, Jakkula and Cook propose an algorithm for prediction enhanced by temporal rules which operates on the output of the Active LeZi predictor. In experiments with small datasets, the prediction accuracy with temporal rules is marginally better than that without temporal rules. It is expected that with larger datasets, the accuracy of the enhanced prediction will see a large improvement [42].

Many prediction algorithms actually fall into machine learning and many new learning algorithms have been proposed

[53], [53]–[57], which can be used to enhance the prediction as well. More recent prediction algorithms have been proposed and they are mostly based on those reviewed above [58]–[62] .

V. Conclusion

The prediction algorithms play key role in providing the intelligence for a smart home by mining the sensed environmental and human activity data. This paper comprehensively reviews prediction algorithms proposed for smart environments in literature. The paper first presents the system models and data that are generally used in prediction algorithms for smart homes. The prediction algorithms are then discussed in details on their features, strengths and weakness. They are also compared and summarized on the models and data types used in prediction. Some enhancements to these algorithms are also reviewed.

VI. Acknowledgement

This material is based on work supported by the National Science Foundation under Grant No. #1408165.

References

[1] Diane J Cook, G Michael Youngblood, Edwin O Heierman III, Karthik Gopalratnam, Sira Rao, Andrey Litvin, and Farhan Khawaja. Mavhome: An agent-based smart home. In *PerCom*, volume 3, pages 521–524, 2003.

[2] Muhammad Raisul Alam, Mamun Bin Ibne Reaz, and Mohd Alauddin Mohd Ali. A review of smart homes—past, present, and future. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1190–1203, 2012.

[3] Mamun Bin Ibne Reaz. Artificial intelligence techniques for advanced smart home implementation. *Acta Technica Corviniensis-Bulletin of Engineering*, 6(2):51, 2013.

[4] Sajal K Das, Diane J Cook, Amiya Battacharya, Edwin O Heierman, and Tze-Yun Lin. The role of prediction algorithms in the mavhome smart home architecture. *IEEE Wireless Communications*, 9(6):77–84, 2002.

[5] Emmanuel Munguia Tapia, Stephen S Intille, and Kent Larson. Activity recognition in the home using simple and ubiquitous sensors. In *International Conference on Pervasive Computing*, pages 158–175. Springer, 2004.

[6] Diane J Cook and Sajal K Das. How smart are our environments? an updated look at the state of the art. *Pervasive and mobile computing*, 3(2):53–73, 2007.

[7] Abhishek Roy, Sajal K Das, and Kalyan Basu. A predictive framework for location-aware resource management in smart homes. *IEEE Transactions on mobile computing*, 6(11):1270–1283, 2007.

[8] Sajal K Das and Diane J Cook. Health monitoring in an agent-based smart home by activity prediction. In *Proceedings of the International Conference on Smart Homes and Health Telematics*, volume 14, pages 3–14, 2004.

[9] Ahmad Lotfi, Caroline Langensiepen, Sawsan M Mahmoud, and M Javad Akhlaghinia. Smart homes for the elderly dementia sufferers: identification and prediction of abnormal behaviour. *Journal of ambient intelligence and humanized computing*, 3(3):205–218, 2012.

[10] Patrice C Roy, Sylvain Giroux, Bruno Bouchard, Abdenour Bouzouane, Clifton Phua, Andrei Tolstikov, and Jit Biswas. A possibilistic approach for activity recognition in smart homes for cognitive assistance to alzheimer's patients. In *Activity Recognition in Pervasive Intelligent Environments*, pages 33–58. Springer, 2011.

[11] Ruijiao Li, Bowen Lu, and Klaus D McDonald-Maier. Cognitive assisted living ambient system: a survey. *Digital Communications and Networks*, 1(4):229–252, 2015.

[12] Shuangquan Wang and Gang Zhou. A review on radio based activity recognition. *Digital Communications and Networks*, 1(1):20–29, 2015.

[13] Lei Yang, Yanyun Ren, and Wenqiang Zhang. 3d depth image analysis for indoor fall detection of elderly people. *Digital Communications and Networks*, 2(1):24–34, 2016.

[14] Ericka Janet Rechy-Ramirez and Huosheng Hu. Bio-signal based control in assistive robots: a survey. *Digital Communications and Networks*, 1(2):85–101, 2015.

[15] Ryan Aipperspach, Elliot Cohen, and John Canny. Modeling human behavior from simple sensors in the home. In *International Conference on Pervasive Computing*, pages 337–348. Springer, 2006.

[16] Sawsan Mahmoud, Ahmad Lotfi, and Caroline Langensiepen. Behavioural pattern identification and prediction in intelligent environments. *Applied Soft Computing*, 13(4):1813–1822, 2013.

[17] Zoubin Ghahramani. An introduction to hidden markov models and bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(01):9–42, 2001.

[18] Jonathan E Cook and Alexander L Wolf. Discovering models of software processes from event-based data. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 7(3):215–249, 1998.

[19] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[20] Hua Si, Yoshihiro Kawahara, Hiroyuki Morikawa, and Tomonori Aoyama. A stochastic approach for creating context-aware services based on context histories in smart home. *COGNITIVE SCIENCE RESEARCH PAPER-UNIVERSITY OF SUSSEX CSRP*, 577:37, 2005.

[21] Sira Panduranga Rao and Diane J Cook. Predicting inhabitant action using action and task models with application to smart homes. *International Journal on Artificial Intelligence Tools*, 13(01):81–99, 2004.

[22] Ehsan Nazerfard and Diane J Cook. Bayesian networks structure learning for activity prediction in smart homes. In *Intelligent Environments (IE), 2012 8th International Conference on*, pages 50–56. IEEE, 2012.

[23] Christian Debes, Andreas Merentitis, Sergey Sukhanov, Maria Niessen, Nikolaos Frangiadakis, and Alexander Bauer. Monitoring activities of daily living in smart homes: Understanding human behavior. *IEEE Signal Processing Magazine*, 33(2):81–94, 2016.

[24] Liming Chen, Chris D Nugent, and Hui Wang. A knowledge-driven approach to activity recognition in smart homes. *IEEE Transactions on Knowledge and Data Engineering*, 24(6):961–974, 2012.

[25] Mirza Adipradhana, IGB Baskara Nugraha, and Suhono Harso Supangkat. Intervention of non-inhabitant activities detection in smart home environment. In *ICT for Smart Society (ICISS), 2013 International Conference on*, pages 1–5. IEEE, 2013.

[26] Shuai Zhang, Sally McClean, Bryan Scotney, Priyanka Chaurasia, and Chris Nugent. Using duration to learn activities of daily living in a smart home environment. In *2010 4th International Conference on Pervasive Computing Technologies for Healthcare*, pages 1–8. IEEE, 2010.

[27] Peng Zhuang and Hao Liang. Energy storage management in smart homes based on resident activity of daily life recognition. In *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 641–646. IEEE, 2015.

[28] Jan Petzold, Faruk Bagci, Wolfgang Trumler, and Theo Ungerer. Next location prediction within a smart office building. *Cognitive Science Research Paper-University of Sussex CSRP*, 577:69, 2005.

[29] Mark Weiser. The computer for the 21st century. *Scientific american*, 265(3):94–104, 1991.

[30] Abhishek Roy, SK Das Bhaumik, Amiya Bhattacharya, Kalyan Basu, Diane J Cook, and Sajal K Das. Location aware resource management in smart homes. In *Pervasive Computing and Communications, 2003.(Per-*

Com 2003). Proceedings of the First IEEE International Conference on, pages 481–488. IEEE, 2003.

[31] Aditi Dixit and Anjali Naik. Use of prediction algorithms in smart homes. *International Journal of Machine Learning and Computing*, 4(2):157, 2014.

[32] Diane J Cook, Manfred Huber, Karthik Gopalratnam, and Michael Youngblood. Learning to control a smart home environment. In *Innovative applications of artificial intelligence*. Citeseer, 2003.

[33] Jacob Ziv and Abraham Lempel. Compression of individual sequences via variable-rate coding. *IEEE transactions on Information Theory*, 24(5):530–536, 1978.

[34] Christina Zeeh. The lempel ziv algorithm. In *URL: http://w3studi. informatik. uni-stuttgart. de/˜ zeehca/Seminar/LempelZivReport. pdf [accessed November 3, 2003]*, 2003.

[35] Karthik Gopalratnam and Diane J Cook. Active lezi: An incremental parsing algorithm for sequential prediction. *International Journal on Artificial Intelligence Tools*, 13(04):917–929, 2004.

[36] Amiya Bhattacharya and Sajal K Das. Lezi-update: an information-theoretic approach to track mobile users in pcs networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 1–12. ACM, 1999.

[37] Hongqing Fang and Jinjin Ruan. An improved position prediction algorithm based on active lezi in smart home. In *Computer Science & Service System (CSSS), 2012 International Conference on*, pages 1733–1736. IEEE, 2012.

[38] Anupam Shukla, Gaurav Ojha, Sachin Acharya, and Shubham Jain. A customized flocking algorithm for swarms of sensors tracking a swarm of targets. *arXiv preprint arXiv:1311.6981*, 2013.

[39] Christoph Moeslinger, Thomas Schmickl, and Karl Crailsheim. A minimalist flocking algorithm for swarm robots. In *European Conference on Artificial Life*, pages 375–382. Springer, 2009.

[40] Jérémy Lapalu, Kevin Bouchard, Abdenour Bouzouane, Bruno Bouchard, and Sylvain Giroux. Unsupervised mining of activities for smart home prediction. *Procedia Computer Science*, 19:503–510, 2013.

[41] Muhammad Raisul Alam, Mamun Bin Ibne Reaz, and MA Mohd Ali. Speed: An inhabitant activity prediction algorithm for smart homes. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 42(4):985–990, 2012.

[42] Vikramaditya R Jakkula and Diane J Cook. Using temporal relations in smart environment data for activity prediction. In *Proceedings of the 24th International conference on machine learning*, pages 20–24, 2007.

[43] Junling Hu and Michael P Wellman. Nash q-learning for general-sum stochastic games. *Journal of machine learning research*, 4(Nov):1039–1069, 2003.

[44] Nirmalya Roy, Abhishek Roy, and Sajal K Das. Context-aware resource management in multi-inhabitant smart homes: A nash h-learning based approach. In *PerCom*, pages 148–158, 2006.

[45] D Schweizer and H Wache. *Learning frequent and periodic usage patterns in smart homes*. PhD thesis, Master's thesis, School of Business, University of Applied Sciences and Arts Nortwestern Switzerland (FHNW) Feb, 2014.

[46] Yongjun Zhang and Tingting Qi. Research on mining association behavior of smart home users based on apriori algorithm. In *International Conference on Materials Engineering and Information Technology Applications*, 2015.

[47] Parisa Rashidi and Diane J Cook. Keeping the resident in the loop: adapting the smart home to the user. *IEEE Transactions on systems, man, and cybernetics-part A: systems and humans*, 39(5):949–959, 2009.

[48] Avinash Achar, Srivatsan Laxman, and PS Sastry. A unified view of the apriori-based algorithms for frequent episode discovery. *Knowledge and information systems*, 31(2):223–250, 2012.

[49] Edwin O Heierman and Diane J Cook. Improving home automation by discovering regularly occurring device usage patterns. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 537–540. IEEE, 2003.

[50] Muhammad Raisul Alam, Md Mamun Ibne Reaz, Mohd Alauddin Mohd Ali, and Salina Abdul Samad. Temporal modeling of human activity in smart homes. *Informacije MIDEM*, 41(2):118–121, 2011.

[51] Vikramaditya R Jakkula and Diane J Cook. Enhancing smart home algorithms using temporal relations. *Technology and Aging*, 21:3–10, 2008.

[52] Vikramaditya Jakkula and Diane J Cook. Learning temporal relations in smart home data. In *Proceedings of the Second International Conference on Technology and Aging, Canada*, volume 33, 2007.

[53] Bin Gu, Victor S Sheng, Zhijie Wang, Derek Ho, Said Osman, and Shuo Li. Incremental learning for ν-support vector regression. *Neural Networks*, 67:140–150, 2015.

[54] Bin Gu, Xingming Sun, and Victor S Sheng. Structural minimax probability machine. 2016.

[55] Bin Gu and Victor S Sheng. A robust regularization path algorithm for ν-support vector classification. 2016.

[56] Zhaoqing Pan, Yun Zhang, and Sam Kwong. Efficient motion and disparity estimation optimization for low complexity multiview video coding. *IEEE Transactions on Broadcasting*, 61(2):166–176, 2015.

[57] Zhangjie Fu, Xinle Wu, Chaowen Guan, Xingming Sun, and Kui Ren. Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement. *IEEE Transactions on Information Forensics and Security*, 11(12):2706–2716, 2016.

[58] M Marufuzzaman, MBI Reaz, Mohd Alauddin Mohd Ali, LF Rahman, et al. A time series based sequence prediction algorithm to detect activities of daily living in smart home. *Methods of information in medicine*, 54(3):262–270, 2015.

[59] Mohamed Tarik Moutacalli, Abdenour Bouzouane, and Bruno Bouchard. Sensors activation time predictions in smart home. In *Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, page 37. ACM, 2015.

[60] Sungjoon Choi, Eunwoo Kim, and Songhwai Oh. Human behavior prediction for smart homes using deep learning. In *2013 IEEE RO-MAN*, pages 173–179. IEEE, 2013.

[61] Serge Thomas Mickala Bourobou and Younghwan Yoo. User activity recognition in smart homes using pattern clustering applied to temporal ann algorithm. *Sensors*, 15(5):11953–11971, 2015.

[62] Jun Huang, Yu Meng, Xuehong Gong, Yanbing Liu, and Qiang Duan. A novel deployment scheme for green internet of things. *IEEE Internet of Things Journal*, 1(2):196–205, 2014.