World Academy of Science, Engineering and Technology
International Journal of Information and Communication Engineering
Vol:12, No:2, 2018

Evaluating Machine Learning Techniques for Activity Classification in Smart Home Environments

Talal Alshammari, Nasser Alshammari, Mohamed Sedky, Chris Howard

International Science Index, Information and Communication Engineering Vol:12, No:2, 2018 waset.org/Publication/10008539

***Abstract*—With the widespread adoption of the Internet-connected devices, and with the prevalence of the Internet of Things (IoT) applications, there is an increased interest in machine learning techniques that can provide useful and interesting services in the smart home domain. The areas that machine learning techniques can help advance are varied and ever-evolving. Classifying smart home inhabitants' Activities of Daily Living (ADLs), is one prominent example. The ability of machine learning technique to find meaningful spatio-temporal relations of high-dimensional data is an important requirement as well. This paper presents a comparative evaluation of state-of-the-art machine learning techniques to classify ADLs in the smart home domain. Forty-two synthetic datasets and two real-world datasets with multiple inhabitants are used to evaluate and compare the performance of the identified machine learning techniques. Our results show significant performance differences between the evaluated techniques. Such as AdaBoost, Cortical Learning Algorithm (CLA), Decision Trees, Hidden Markov Model (HMM), Multi-layer Perceptron (MLP), Structured Perceptron and Support Vector Machines (SVM). Overall, neural network based techniques have shown superiority over the other tested techniques.

***Keywords*—Activities of daily living, classification, internet of things, machine learning, smart home.

I. Introduction

In the recent years, we have seen a rapid increase of the Internet of Things (IoT) applications, such as smart homes. These environments generate huge amount of sensory data which has the potential to allow stakeholders, i.e. home owners, to analyse this generated data for monitoring, detection and classification of activities in order to make timely decisions. Using machine learning to learn and predict the inhabitants' activities is becoming an interesting and active research area. Home automation has existed for some time; now the technology is at a stage where individual households can make use of it. Home automation provides comfort, home energy management and security and can help the elderly and disabled to receive quality care. There has been proposed designs and implementations of interactive systems which provide citizen awareness of resource use [1]. These propositions provide simple and efficient management of a house system to enhance daily activities. Although this approach has great potential to help residents with sustainable living, there are many challenges as to how the technology is integrated with the smart home environment.

Talal Alshammari is with Staffordshire University, College Road, ST4 2DE Stoke-on-Trent, UK (e-mail: talal.alshammari@research.staffs.ac.uk).

Nasser Alshammari is with Staffordshire University, College Road, ST4 2DE Stoke-on-Trent, UK (e-mail: nasser.alshammari@research.staffs.ac.uk).

Mohamed Sedky is with Staffordshire University, College Road, ST4 2DE Stoke-on-Trent, UK (e-mail: m.h.sedky@staffs.ac.uk).

Chris Howard is with Staffordshire University, College Road, ST4 2DE Stoke-on-Trent, UK (e-mail: c.howard@staffs.ac.uk).

Machine learning techniques have been heavily applied to smart home environments, however up to the knowledge of the authors, there is a lack of comprehensive evaluation of such machine learning algorithms in this domain. The contribution of this paper is twofold: 1. a review of existing research in classification of ADLs, 2. a comprehensive evaluation of state-of-the-art machine learning techniques application in the context of smart homes.

There are many smart home research efforts focusing on using machine learning techniques in a domestic environment. However, the current machine learning techniques are lacking in certain aspects. Some algorithms in the smart home domain are subject to error when predicting the inhabitant's behaviour. When multiple inhabitants are living in the same home, multi-class classification is needed, which is a challenging and difficult task to perform due to the nature of the data. Moreover, the data readings from sensors are always noisy and subject to many uncertain variables such as missing data and faulty sensors. Identifying and learning spatio-temporal relationships between the sensors' readings to achieve high accuracy classification is needed [2], [3].

II. Machine Learning Techniques for ADLs Classification

Machine learning has been widely applied to develop probabilistic and statistical methods and sequence-learning algorithms to predict activities of daily living (ADLs) of inhabitants. Machine learning techniques can be divided into three categories based on the availability of labelled datasets, as such:

- **Unsupervised** techniques are used when there are no ground truth labels available for the ADLs. In this situation, clustering techniques are used to group similar ADLs into clusters. However, clustering techniques alone do not classify and predict ADLs. They are usually used with other techniques to facilitate certain aspects of the learning model, such as performing a pre-processing step of the data.
- **Semi-supervised** techniques are used when parts of the ground truth labels are available. This is usually the case in real-world datasets because the inhabitants are asked to record their activities manually. This approach is prone to human errors.
- **Supervised** techniques are used when there is full availability of the ground truth labels. Which is mostly the case in synthetic datasets that are generated using simulation tools [4]-[6].

One of the projects that used supervised techniques is MavHome which aims at creating a smart home environment that acts as an intelligent agent by reading the sensors' data and intelligently manipulating the environment using device controllers. The project proposes a scalable architecture to achieve their goals. Each agent in MavHome architecture is composed of four layers, which are the decision layer, the information layer, the communication layer, and the physical layer. The relevant layer to the research at hand is the decision layer, which is responsible for deciding the agent's action based on the gathered information. The project uses a Smart Home Inhabitant Prediction (SHIP) algorithm which works by searching and matching recent sequence of events with previously captured sequences. To evaluate the proposed algorithm, its performance was tested on a real dataset and it scored a classification accuracy of 53.4% and 94.4% on a synthetic dataset. The simplicity of the SHIP algorithm is one of its strengths. However, it has a limitation of not being able to operate in an online fashion. The whole historical activities must be stored and processed offline. To overcome this limitation, the project developed Active LeZi (ALZ), which is a sequential prediction algorithm. They tested this algorithm performance on synthetic dataset and it scored 87% accuracy. Moreover, the project applied a Task-based Markov Model (TMM) and it scored 74% accuracy on a 30-day synthetic dataset [7].

A. Support Vector Machines (SVMs)

Support Vector Machines (SVMs) are supervised learning models used for classification and regression analysis [8]. Also, SVMs have many benefits, high dimensional feature space. SVMs are effective when the number of samples is less than the number of the dimensions in the dataset. Moreover, SVMs can be efficient on memory usage. SVMs can be used with different kernel functions which will allow the model to learn complex decision function. On the other side, SVMs have some disadvantages such as over-fitting, which can occur when the number of features is greater than the number of samples [9].

SVMs have been used in the literature for classification of ADLs in the health domain. Health Smart Home is one of these efforts which includes real data collected from various sensors, to evaluate the SVM algorithm. They installed many sensors and microphones in the environment and obtained a classification accuracy of 75% and 86% using a polynomial kernel and a Gaussian kernel respectively [10].

B. Hidden Markov Model (HMM)

Hidden Markov Model (HMM) is an unsupervised generative probabilistic model. The HMM deals with hidden states, which means the state is not observed directly. The transition from one hidden state to another can be modeled as a Markov process. The HMM is suitable for sequential datasets. States have a probability distribution on the likely output symbol [11].

The use of HMM was proposed to classify ADLs of smart home multi-inhabitants. The hidden states were modeled to be the activities' labels and the observations are the sensors' readings. To evaluate the accuracy of the model, ARAS dataset was created [12]. The dataset represents real-world activities captured from multi-residents in two real houses. HMM average accuracy was 61.5% in house A, while house B the average accuracy approached 76.2%.

C. Decision Trees (DT)

Decision Trees (DT) are supervised non-parametric learning models used for classification and regression. Non-parametric learning models do not assume that a probability distribution generated the data. A DT model learns simple condition rules inferred from the labelled data. Thus, a DT model is easy to interpret and understand. Moreover, the model usually offers good performance and its time complexity is low. Some machine learning algorithms cannot work with certain data types. However, Decision Trees are able to work with categorical and numerical datasets alike. For multi-label classification problems, DT can work and offer good solutions. Very little data pre-processing is required when creating these models. However, DT suffer from some disadvantages. DT cannot work with missing values without preparing the dataset. They are prone to over-fitting and can produce complex models that are sensitive to small changes in the dataset, which do not generalise well and may produce unstable models. There are some datasets that can be hard for the DT to learn. Moreover, it is not guaranteed that the learned DT is the optimal tree [9].

DT can be used to classify ADLs of smart home inhabitants. E-ID5R is an extension of the DT algorithm to allow it to work with multi-label classification problems [13]. The accuracy of E-ID5R was evaluated using the same ARAS dataset [12]. E-ID5R classification accuracy approached 40% on house A and 82% on house B.

D. Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (SGD) is an iterative algorithm used to find the minimum and maximum value of a function. Usually it is used with convex loss function to find the minimum error. It can be used with linear classifiers, such as SVMs, for classification and multi-label classification problems. It is able to work and scale with large datasets. However, it needs several hyper-parameters to be set, such as the learning rate and the number of iterations [9].

The logistic regression with SGD algorithm were applied in order to develop a scalable diagnosis model for health care applications. To assess the proposed algorithm, they used the Cleveland Heart Disease Database (CHDD) which collected data from wearable body sensors used to measure the blood pressure and heart disease rate. Logistic regression with SGD algorithm enabled the model to predict and classify the heart disease status. The accuracy of training and validation on the data sample was 81.99% and 81.52% respectively [14].

E. AdaBoost

AdaBoost is a supervised learning algorithm used for classification and regression. The algorithm uses a group of weak learners or weak prediction models. The final prediction is the result of all the predictions from the weak learners combined. Thus, the model can be thought of as a majority voting system. AdaBoost can be used for multi-label classification problems as well [15], [9].

AdaBoost algorithm was proposed to deal with the classification of eating and meal preparation in the smart home. In order to reduce the number of sensors and focus on using the main sensors required for this task, they used the dataset from the MIT PlaceLab project. The authors used only 8 sensors from over 300 sensors, obtaining a rate accuracy of 82% [16].

F. Hierarchal Temporal Memory (HTM)

The HTM theory attempts to model the architecture and structure of the neocortex, the front part of a human brain. The focus of the theory is on the neocortex because it is envisaged to be where the human intelligence resides. The cortical learning algorithm (CLA) is a machine learning algorithm that is based on HTM theory, which aims to explain the structural and algorithmic characteristics of the neocortex [17].

A typical CLA model will consist of multiple regions. The first region is the encoding region which can contain one or more encoders that read the input data and convert it to Sparse Distributed Representation (SDRs). The next region is the Spatial Pooler (SP), which receives the outputted SDRs from the encoding region below. The Spatial Pooler will learn the spacial features of the passed SDRs and create another SDRs and output it to the next region. The third region is a Temporal Memory region (TM) which learns the temporal changes in the SDRs. Finally, for prediction and classification problems, a CLA/SDR Classifier region sets at the top of the HTM model. The CLA/SDR Classifier region decodes the state of the HTM model and produces predictions [18].

The HTM theory and its algorithmic implementation, the CLA, have been applied in many domains. Such as vision [19], [20], natural language processing (NLP) [21], and anomaly detection in smart homes [22].

The CLA was used to classify “healthy” and “sick” patients using a dataset that contains 70 patients. The dataset captures Electrocardiography signals (ECG). The CLA performance was slightly better than the multi-layer neural network [22].

G. Multi-Layer Perceptron (MLP)

A Multi-layer Perceptron (MLP) is a feed-forward artificial neural network model, it maps values of input data onto a value of suitable output. It contains multiple hidden layers which are between the input and output layers. Each node is a neuron and every layer is fully connected to the next layer via weights. For each neuron, a weighted sum is calculated from the previous layer and then the result is passed to an activation function. After applying the activation function, the result is passed to the next layer. There are several types of activation functions, such as the sigmoid function, the hyperbolic tangent function and the softmax function. The MLP model uses back propagation for training the network in order to reduce the error [23]. The structured perceptron is an extension of the standard perceptron that can predict structured data and usually it is used with an inference algorithm, such as the Viterbi algorithm [24], [25].

The Back Propagation Neural Network (BPNN) has been applied to classify ADLs in a smart home. To assess the proposed algorithm, they used The Centre for Advanced Studies in Adaptive Systems (CASAS) that is a project for creating real smart homes for the researchers in this field. They demonstrated that the size of the neurons play important role to reduce the error rate [26].

H. Long Short Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a Recurrent Neural Network (RNN). The LSTM model is good for classifying and predicting sequences, such as recognition of speech and handwriting. In regular RNNs, it is hard to train the model when the dependency of prediction has been seen a long time ago. This problem is known as the ”long-term dependency problem” [27]. LSTM is an extension of RNNs to overcome this problem [28].

Deep convolutional and LSTM units framework was proposed in the domain of Human Activity Recognition (HAR) [29]. It was used deep convolutional to extract special features from sensors data and LSTM to model temporal dynamics. The proposed framework was validated on two datasets, Opportunity dataset and Skoda dataset [30], [31]. The framework obtained 96% F1 score on the Skoda dataset and 93% F1 score on the Opportunity dataset.

I. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) work like other neural networks but have a different construction. They are usually used to recognize visual patterns through images and videos. CNN can work with image data without requiring pre-processing for the data [32]. CNNs were applied in different domains such as natural language processing (NLP) and recommender systems.

CNN has been proposed to recognize the inhabitants’ ADLs. The CNNs were able to capture local dependency of the activities and showed good scale invariance. Three datasets (Skoda, Opportunity, Actitracker) were used validate the proposed technique. The technique accuracy is 88%, 77%, and 97% on Skoda, Opportunity, and Actitracker respectively [33].

III. Evaluation Methodology

The evaluation of the machine learning algorithms is limited by a lack of standard real datasets from smart homes. Due to the high cost of building real smart home datasets, there is a need for powerful simulation tools that can represent the ADLs of the inhabitants. These simulation tools offer flexibility, scalability and accessibility for the researcher [34]-[37].

To evaluate the performance, this research will use real and simulated smart home datasets. The following subsections describe the preparation of the datasets and the performance metrics used to assess the quality of the models.

A. Real Smart Home Dataset

For the real-world dataset, ARAS dataset was used [12]. The dataset was generated from real activities engaged by multi-inhabitants in two separate smart home during a period spanning two months. The dataset contains 20 columns of binary data that represent the sensor values (0 represents OFF state, and 1 represents ON state) sampled each second; whereby, column 21 represents the activity labels for Resident 1, and column 22 represents the activity labels for Resident 2. The residents perform various labeled activities (27 different activities to be exact).

This research used this dataset but has combined the data into five activities, namely: Other, Sleeping, Eating, Personal, and Relaxing. Resulting in a total collection of 25 activities. Another pre-processing step was performed to reduce the size of the dataset from 86400 records per day to 1440 records. The reduction was done by changing the sample rate from seconds pre day to minutes per day. The reduction took care of the sensors' values that changed in between minutes by retaining these changes. This step reduced the overall dataset size from 5184000 records to 86400 records. The motivation behind this step came after the analysis of the dataset which revealed that most of the transitions of activities take longer than one minute.

B. Synthetic Smart Home Dataset

This research used OpenSHS [34] which is an open source simulation tool that offered the flexibility needed to generate the inhabitant's data for classification of ADLs. OpenSHS was used to generate several synthetic datasets that includes 29 columns of binary data representing the sensor values. The sampling was done every second. Seven participants were asked to perform their simulations using OpenSHS. Each participant generated six datasets resulting in forty-two datasets in total. The participants self-labelled their activities during the simulation. The labels used by the participants were: Personal, Sleep, Eat, Leisure, Work, Other. The total number of records is 2674910 records.

C. Experiment Design

After the preparation of the forty-two OpenSHS synthetic datasets and the two real-world datasets, the records of each dataset are fed to a machine learning model. For each dataset, the data was split into two parts: a training part, where the model is learning from the data without scoring its performance, and a testing part, where the model's classification accuracy is evaluated. The training part size is 80% of the total size of the dataset and the remaining 20% is used for testing. No shuffling was performed because the activities have natural sequence progression. All of the sensors readings were fed to the tested models with the exception of the timestamp column. We did not include the timestamp column as one of the input features because of the inability for some of the models to work with this data type and to ensure a levelled playing ground for all the evaluated models.

D. Performance Metrics

The inhabitants' ADLs usually vary from one inhabitant to another. Moreover, the ratio of the performed activities for each inhabitant is usually not similar. For example, the "Sleeping" activity may constitute a bigger portion of the whole activity space. Thus, it is good to assume that the labels are imbalanced [29]. Therefore, simple metrics such as the accuracy metric, shown in (1), are not suitable for this type of datasets.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

A naive classifier that predicts the labels with highest frequencies could get high accuracy score. To overcome this issue, F1 score was used. As shown in (2), F1 score is defined in terms of precision and recall.

$$F1 = 2 \times \frac{(precision \times recall)}{(precision + recall)} \tag{2}$$

The precision is the ratio of the relevant points that have been selected by the model to the total selected points, as shown in (3), where *TP* is the *True Positives* and *FP* is the *False Positives*.

$$precision = \frac{TP}{TP + FP} \tag{3}$$

The recall is the ratio of relevant points that have been selected by the model to the overall total of the relevant points, as show in (4), where *TP* is the *True Positives* and *FN* is the *False Negatives*.

$$recall = \frac{TP}{TP + FN} \tag{4}$$

IV. Results and Discussion

The results of the selected machine learning models are obtained using several software packages. Primarily we used scikit-learn [9] and keras [38]. Due to the various configurations used for certain models, we abbreviated the models as shown in Table I. Using Precision, Recall, and F1-measure as evaluation metrics, the results are shown in Fig. 2 for House A from ARAS dataset, Fig. 3 for House B from the same dataset, and Fig. 4 from the OpenSHS synthetic dataset.

The choice of the kernel type used with SVM proved to be crucial. Using a polynomial kernel produced the worst results especially in House A. While using a linear kernel and an RBF kernel produced the best results for the SVM algorithm.

Increasing the number of estimators for AdaBoost did not improve the results and the best results obtained with ten estimators.

In another research effort, the HMM was evaluated on the ARAS dataset and the reported accuracy results for House A is 61.5% and for House B is 76.2% [12]. Ours obtained accuracy results of 53.9% for House A and 92.3% for House B. It is worth noting that the differences in the results could be attributed to the preformed pre-processing steps. The previously mentioned work used leave-one-out cross validation

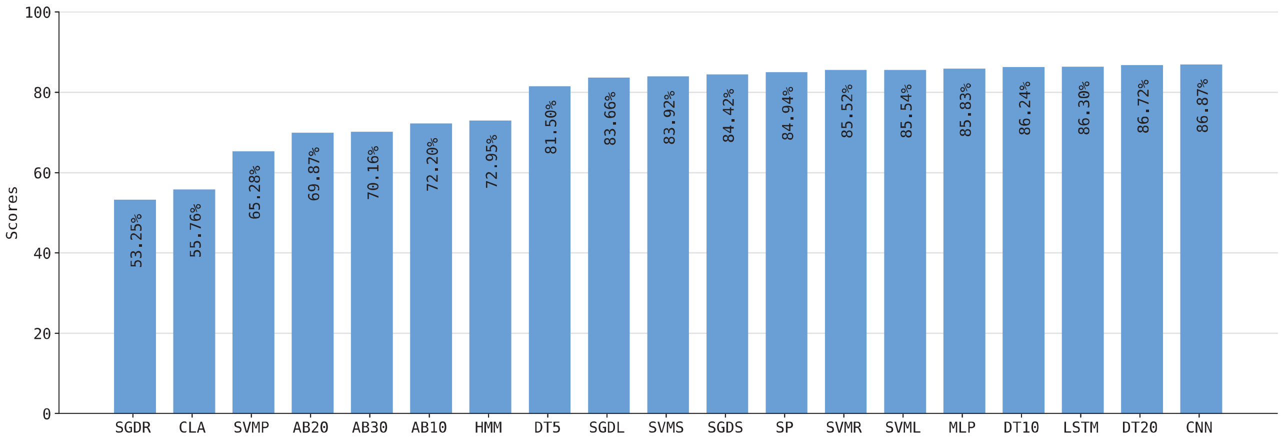

Fig. 1 The average F-measure score across all datasets

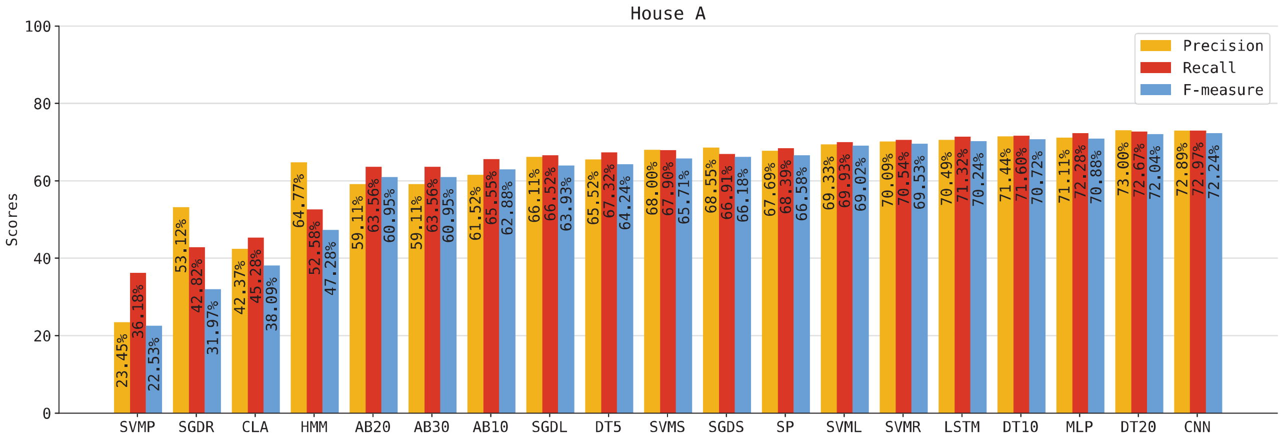

Fig. 2 Results of House A dataset

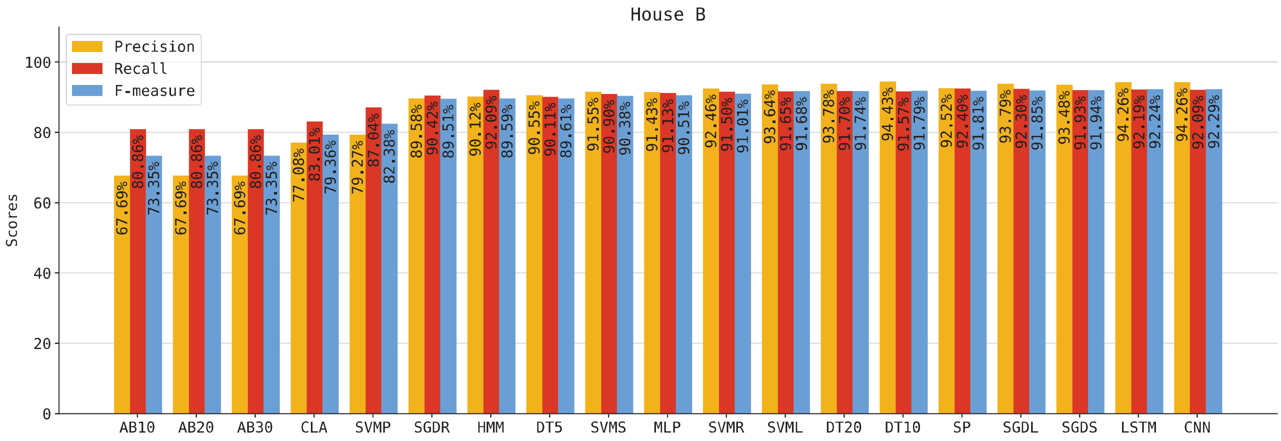

Fig. 3 Results of House B dataset

on the dataset and the reported results were the average accuracy. For House A, the minimum accuracy is 46.3% and the maximum accuracy is 88.4%. For House B, the minimum accuracy is 31.1% and the maximum accuracy is 96.7%. In our work we did not use any cross-validation technique because we assume the activities have natural sequence progression and any cross-validation step will break this assumption.

As expected, the obtained results for the real datasets were more challenging for the models to learn than the synthetic datasets.

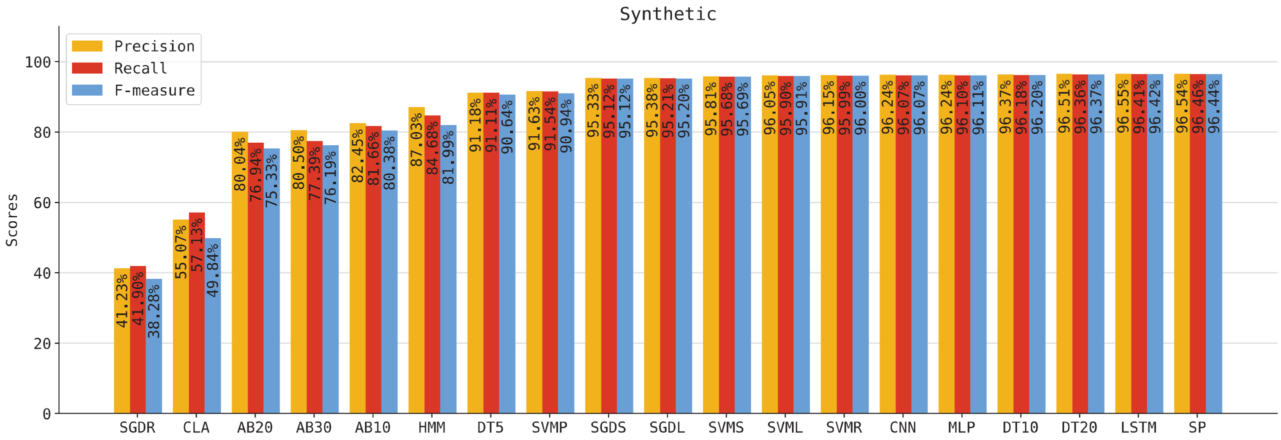

Fig. 4 Results of OpenSHS Synthetic dataset

TABLE I
ABBREVIATIONS FOR THE EVALUATED MODELS

AB10	AdaBoost with 10 estimators
AB20	AdaBoost with 20 estimators
AB30	AdaBoost with 30 estimators
CLA	Cortical Learning Algorithm
CNN	Convolutional Neural Network
DT5	Decision Tree with max depth of 5
DT10	Decision Tree with max depth of 10
DT20	Decision Tree with max depth of 20
HMM	Hidden Markov Model
LSTM	Long Short Term Memory
MLP	Multi-layer Perceptron
SVMR	Support Vector Machine with RBF kernel
SVML	Support Vector Machine with linear kernel
SVMP	Support Vector Machine with polynomial kernel
SVMS	Support Vector Machine with sigmoid kernel
SGDS	Stochastic Gradient Descent with linear SVM function
SGDR	Stochastic Gradient Descent with regression function
SGDL	Stochastic Gradient Descent with logistic regression function
SP	Structured Perceptron

The performance of the CLA algorithm could have been improved if a custom and more suitable encoder was used. The high-dimensional and binary nature of the data types in the datasets, was a challenge for the existing encoders. The existing encoders are designed to work with simple scalar and categorical datatypes.

The overall performance of all the evaluated machine learning algorithms across all datasets using F-measure score is summarised in Fig. 1. The results show competitive performance of the evaluated algorithms. However, three of the top five algorithms are based on neural networks.

V. CONCLUSION

Classification plays an important role in the field of artificial intelligence and machine learning for creating smart systems that are able to make decisions more reliably [39]. Learning from historical events and attempting to predict future events is an essential requirement for smart homes. In this paper, several state-of-the-art machine learning algorithms were evaluated on real-world datasets and OpenSHS synthetic datasets of smart homes. The algorithms based on neural networks showed superior performance over other algorithms. DT, LSTM, SVM and SGD are good candidates for the task at hand. However HMM, AdaBoost and CLA showed much less performance accuracy. HMM as an unsupervised algorithm does not compete well with other supervised algorithms in this context. The CLA results indicate that there is a need for a custom encoder that is more suitable for the nature of smart home datasets.

ACKNOWLEDGEMENT

Talal Alshammari and Nasser Alshammari are carrying out their Ph.D. studies at Staffordshire University. The Ministry of Education in Saudi Arabia funds and supports their research projects.

REFERENCES

[1] L. Bartram, J. Rodgers, and R. Woodbury, "Smart homes or smart occupants? supporting aware living in the home," *Human-Computer Interaction–INTERACT 2011*, pp. 52–64, 2011.

[2] B. Minor, J. R. Doppa, and D. J. Cook, "Data-driven activity prediction: Algorithms, evaluation methodology, and applications," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 805–814.

[3] I. Fatima, M. Fahim, Y.-K. Lee, and S. Lee, "Effects of smart home dataset characteristics on classifiers performance for human activity recognition," in *Computer Science and its Applications*. Springer, 2012, pp. 271–281.

[4] E. M. Tapia, S. S. Intille, and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," in *Pervasive*, vol. 4. Springer, 2004, pp. 158–175.

[5] E. Thomaz, T. Plötz, I. Essa, and G. D. Abowd, "Interactive techniques for labeling activities of daily living to assist machine learning," in *Proceedings of Workshop on Interactive Systems in Healthcare*, 2011.

[6] S. T. M. Bourobou and Y. Yoo, "User activity recognition in smart homes using pattern clustering applied to temporal ann algorithm," *Sensors*, vol. 15, no. 5, pp. 11 953–11 971, 2015.

[7] D. J. Cook, M. Youngblood, E. O. Heierman, K. Gopalratnam, S. Rao, A. Litvin, and F. Khawaja, "Mavhome: An agent-based smart home," in *Pervasive Computing and Communications, 2003.(PerCom 2003). Proceedings of the First IEEE International Conference on*. IEEE, 2003, pp. 521–524.

[8] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[10] A. Fleury, M. Vacher, and N. Noury, "Svm-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results," *IEEE transactions on information technology in biomedicine*, vol. 14, no. 2, pp. 274–283, 2010.
[11] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state markov chains," *The annals of mathematical statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.
[12] H. Alemdar, H. Ertan, O. D. Incel, and C. Ersoy, "Aras human activity datasets in multiple homes with multiple residents," in *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2013, pp. 232–235.
[13] M. Prossegger and A. Bouchachia, "Multi-resident activity recognition using incremental decision trees," in *Adaptive and Intelligent Systems*. Springer, 2014, pp. 182–191.
[14] G. Manogaran and D. Lopez, "Health data analytics using scalable logistic regression with stochastic gradient descent," *International Journal of Advanced Intelligence Paradigms*, vol. 8, no. 2, 2017.
[15] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *European conference on computational learning theory*. Springer, 1995, pp. 23–37.
[16] B. Logan and J. Healey, "Sensors to detect the activities of daily living," in *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*. IEEE, 2006, pp. 5362–5365.
[17] J. Hawkins and S. Ahmad, "Why neurons have thousands of synapses, a theory of sequence memory in neocortex," *Frontiers in Neural Circuits*, vol. 10, p. 23, 2016. [Online]. Available: https://www.frontiersin.org/article/10.3389/fncir.2016.00023.
[18] J. Hawkins, S. Ahmad, S. Purdy, and A. Lavin, "Biological and machine intelligence (bami)," 2016, initial online release 0.4. (Online). Available: http://numenta.com/biological-and-machine-intelligence/.
[19] R. Škoviera and I. Bajla, "Image classification based on hierarchical temporal memory and color features," *Slovak Academy of Sciences, MEASUREMENT*, 2013.
[20] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning-a new frontier in artificial intelligence research (research frontier)," *IEEE computational intelligence magazine*, vol. 5, no. 4, pp. 13–18, 2010.
[21] F. D. S. Webber, "Semantic folding theory and its application in semantic fingerprinting," *arXiv preprint arXiv:1511.08855*, 2015.
[22] M. Otahal and O. Stepankova, "Anomaly detection with cortical learning algorithm for smart homes," *SMART HOMES*, vol. 20144, p. 24.
[23] F. Rosenblatt, "Principles of neurodynamics. perceptrons and the theory of brain mechanisms," Cornell Aeronautical Lab Inc Buffalo NY, Tech. Rep., 1961.
[24] M. Collins, "Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002, pp. 1–8.
[25] L. Zhu, Y. Chen, X. Ye, and A. Yuille, "Structure-perceptron learning of a hierarchical log-linear model," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.
[26] H. Fang and L. He, "Bp neural network for human activity recognition in smart home," in *Computer Science & Service System (CSSS), 2012 International Conference on*. IEEE, 2012, pp. 1034–1037.
[27] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.
[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[29] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.
[30] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha *et al.*, "Collecting complex activity datasets in highly rich networked sensor environments," in *Networked Sensing Systems (INSS), 2010 Seventh International Conference on*. IEEE, 2010, pp. 233–240.
[31] P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Troster, "Activity recognition from on-body sensors: accuracy-power trade-off by dynamic sensor selection," *Lecture Notes in Computer Science*, vol. 4913, p. 17, 2008.
[32] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[33] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors," in *Mobile Computing, Applications and Services (MobiCASE), 2014 6th International Conference on*. IEEE, 2014, pp. 197–205.
[34] N. Alshammari, T. Alshammari, M. Sedky, J. Champion, and C. Bauer, "Openshs: Open smart home simulator," *Sensors*, vol. 17, no. 5, 2017.
[35] D. Cook, M. Schmitter-Edgecombe, A. Crandall, C. Sanders, and B. Thomas, "Collecting and disseminating smart home sensor data in the casas project," in *Proceedings of the CHI Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research*, 2009, pp. 1–7.
[36] K. Bouchard, A. Ajroud, B. Bouchard, and A. Bouzouane, "Simact: a 3d open source smart home simulator for activity recognition," *Advances in Computer Science and Information Technology*, pp. 524–533, 2010.
[37] J. Synnott, C. Nugent, and P. Jeffers, "Simulation of smart home activity datasets," *Sensors*, vol. 15, no. 6, pp. 14 162–14 179, 2015.
[38] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.
[39] K. Gopalratnam and D. J. Cook, "Active lezi: An incremental parsing algorithm for sequential prediction," *International Journal on Artificial Intelligence Tools*, vol. 13, no. 04, pp. 917–929, 2004.