mühendislikdergisi
Dicle Üniversitesi Mühendislik Fakültesi

Dicle University Journal of Engineering (DUJE)

web: http://dergipark.gov.tr/dumf

Research Article

Artificial Intelligence Helps Protect Smart Homes against Thieves

Zeydin Pala [1,*], Orhan Özkan [2]

[1] Muş Alparslan University Faculty of Architecture and Engineering,
Department of Computer Engineering, Muş.Turkey, https://orcid.org/0000-0002-2642-7788
[2] YYU, Institute of Science, Department of Artificial Intelligence and Robotics, Van, Turkey, https://orcid.org/0000-0001-9502-8222

ARTICLE INFO

Article history:

Received 7 March 2020
Received in revised form 5 May 2020
Accepted 11 May 2020
Available online 30 September 2020

Keywords:

Artificial intelligence, machine learning, classification, smart home security, thefts

Doi: 10.24012/dumf.700311

ABSTRACT

Interaction with the environments in which humans live is increasing more and more, and Artificial Intelligence (AI) offers significant contributions to this. Although the topic of smart homes has attracted a great deal of attention from researchers, the AI-based application in this area is still in its infancy. In this study, a home security automation system, which is quite simple, but smart and AI-based, is proposed. When the home-dwellers were not at home, the home lighting system tried to be managed with AI at night, as if life was still there. The AI-based smart home physical design was done using Arduino equipment and was tried to be adapted to the real-life environment with software support. As if there was someone at home, a special dataset, which was consisted of nine inputs, one output vector and about 5500 samples was created to turn on/off the home lights in a manner suitable for night life. The home lighting system was successfully managed using an AI-based system that learns nightlife lighting habits.

The proposed system performance was tested in support of commonly used machine learning classification algorithms such as Multi-layer perceptron (MLP), Linear support vector machine (L-SVM), Gaussian Naive Bayes (NB), and linear discriminant analysis (LDA). The accuracy values of MLP, L-SVM and NB algorithms were 96.69%, 94.98% and 91.23%, respectively. Our results show that a home with AI could be safer and more secure against theft.

* Corresponding author
Zeydin, Pala
✉ z.pala@alparslan.edu.tr

Please cite this article in press as Z. Pala, O. Ozkan, "Artificial Intelligence Helps Protect Smart Homes against Thieves", DUJE, vol. 11, no.3, pp. 945-952, September 2020.

Introduction

Homes where people spend majority of their lives have changed over the years with developing technology and emerging ideas and demands thereby resulting in the concept of smart homes. More specifically, smart homes are defined as homes that utilize computer technology for providing many conveniences that are not readily available in classical homes; or to facilitate routine tasks thereby improving the comfort levels of residents.

Homes that are remotely accessed by many people with features such as lamps, boilers, plugs, etc. that can also be turned on and off remotely are described as smart homes. Researchers describe such homes as smart homes which are sensitive to the needs of individuals and provide them with advanced solutions such as learning algorithms in order to automatically learn the behaviors of individuals in addition to controlling systems such as heating and lighting [1].

AI techniques such as artificial neural networks (ANN), multi-agent systems, statistical methods, data compression methods, and fuzzy logic are widely used in contemporary smart homes [2]. Artificial intelligence (AI) techniques can be used to automatically control lighting and heating systems in smart homes as well as to predict and automatically perform the next user action [3]. Smart homes comprise a popular subject with regard to academic work and many studies are conducted on this topic.

Homes provide control and greater comfort and quality of life than those in the past as they get smarter and smarter with the passing of time. However, it is still not possible to talk about homes that are completely safe with the smart home concept.

Today, because of pandemic diseases such as Covid-19, smart homes will become more and more smart and their importance to protect human health will be presented in the home environment. For example, issues such as disinfecting people entering the house, ventilating the house automatically from time to time, and keeping the oxygen rate of the house at desired values are the first features that come to mind.

Having many remotely operated features is not enough to prevent theft. Homeowners who go on holiday are generally discomforted due to thoughts on the possibility that their homes might be robbed any moment. This may turn into a nightmare.

Although the smart homes and its technologies are getting safer, there are always malicious persons whose aim are to cause harm to the smart home, steal significant thing without permission.

How can homes be safer when the residents are outside? Can AI help smart homes protect themselves against thieves during holidays? The answer to these questions are also the subject of this study. Statistics puts forth that homes which are robbed at night constitute a larger majority.

Life at home was evaluated in two different ways in this study which are namely day life and night life. The algorithms that model night life and day life are based on two different scenarios. The reason for working with such clearer expressions is to induce an impression by way of AI that the homeowners are not at home or on the contrary that daily life is still ongoing in the home as if all the residents are still there. A dynamic environment was presented in the study with home lights turned on and off depending on the normal life style for night life.

There are studies on the different uses of AI or machine learning (ML) algorithms for smart homes and for building automation in literature. However, no work has been found which provides protection against theft by using these algorithms on an automation application and by utilizing AI. It is considered that the study will provide significant contributions to the literature on the use of AI techniques against theft in smart homes.

This manuscript is designed as follows: In Chapter 2, related works are presented. In Chapter 3, the general structure and functioning of the system, dataset creation and background of the study are discussed. In section 4, our results are analyzed and comparisons of the other studies are presented. Conclusions and future work presented in Section 5.

Related work

There are many studies in the literature about AI and ML. However, we tried to compile some of those related to our work here. Gariba and

Pipaliya in [4] proposed a framework for a smart home energy management system with the applications of ML algorithms to mitigate energy consumption by modeling human behavior. In the study, they used the combination of Hidden Markov Model (HMM) and Naive Bayes (NB) classification algorithms.

In their work [5] Dixit and Naik used to model the movements of people living in a home and try to predict their next movements with the help of ML algorithms. In [6], they used ML and studied the use of sound to detect fall of elderly events.

In their work [7], the authors attempted to classify the movements of workers in a smart home with ML algorithms. ANNs have shown better performance than many algorithms used in the study.

The authors in [8] attempted to predict the behavior of people living in a smart home environment by using ML algorithms comparatively.

In [9], the authors introduced an activity-aware approach by using center for advanced studies in adaptive systems and anomaly detection algorithm to security monitoring and threat detection and take appropriate action in smart home environment.

The authors in [10] have been trying to identify activities that are specific to a particular location in a home environment by training supervised learning algorithms with Opportunity dataset. They used sensors to collect information about the location.

The authors in [11] compared the results of Convolutional neural networks (CNN), long short term memory (LSTM), Naive Bayes (NB), Hidden Markov models (HMM), Hidden Semi-Markov models (HSMM) and Conditional random fields (CRF) to predict activity in a smart environment. The authors stated in their main results that 1D-CNN and LSTM algorithms are similar but perform better than others. The authors in [20] used their own synthetic datasets consisting of 584 records and 6 properties in their work.

The authors [21] created their own synthetic datasets using the Emotiv Epoc BCI headset with 14 EEG channels. The recording process is 128 Hz. carried out by sampling.

In the literature, no study has been made to model the lighting system as if it were in real life against thefts, even though the actions of people based on ML in the home were attempted. This work in this way is originally contributing to the literature.

General architecture of the system

The main backbone of this study is based on hardware and software components. A special home prototype was designed to represent the life going on in a real home. Arduino mega and 8-channel relay was used among the hardware module components that control the home lights. On the software side, an AI module controlling the hardware module was developed and used. While Arduino mega is used to control the light system of the prototype home based on AI, the 8-channel relay was used to switch a load operating at high power or current with the electromagnetic field created by the low level current.

The smart home included in this study is comprised of a standard hall, three rooms, a kitchen and a bathroom-toilet. Certain parts of the home are lit up at certain times of the night and sometimes the lights of all rooms are left on for a certain time interval, which are turned off afterwards.

Lights are on or off at certain times in certain rooms of any regular home environment where people live.

At first, all light related tasks are recorded by the system during the night after which the dataset to be used for ML is generated.

It is more logical and realistic to turn on/off home lights in accordance with a scenario taken from real life rather than doing it randomly.

As shown in Figure 1, Arduino mega was used for home data collection in this study, whereas mySQL database was used for registering home data, C# programming language for managing the processes in general and the Python programming language for machine learning.

Python was selected for this study as one of the most suitable languages for data intelligence. Moreover it is more powerful, fast and easier to use when compared with languages such as C++, Java, Julia, R, and MATLAB even though they are used in data science. Furthermore, an open-source language is freely distributed.

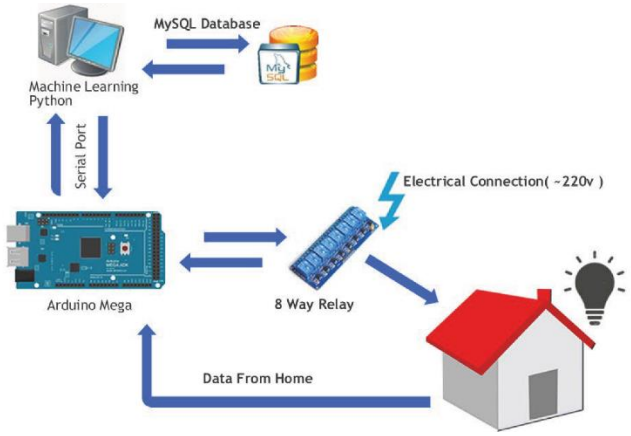

Figure 1. General system architecture

Creating dataset for machine learning in home automation

Data sets are required to train algorithms used in ML. After completing the hardware designs of the study, a dataset was created based on the habits of turning on/off the lights as if it was a regular home environment where normal life is still ongoing. Since there was no existing dataset for the smart home systems, the required data for the system under consideration was generated in accordance with the actual home life. The dataset was generated for the time period between 19:00 and 23:59 when the lights were generally on. It is assumed that the lights are off during the remaining time interval, except hall.

As shown in Table 1, information on lighting level in the home generated in accordance with real life are stored in the dataset under titles of hour (H), minute (M), salon (S), kitchen (K), nursery (N), living room (L), bedroom (B), bath-wc (B-WC), hallway (HW) and lighting level (LL).

Table 1. A sample dataset fragment with 8 samples showing classification based on lighting level

H	M	S	K	N	L	B	B-WC	HW	LL
19	50	0	0	1	0	0	0	0	1
20	32	1	1	01	0	0	0	0	2
20	52	1	1	0	0	0	0	0	2
22	22	1	0	0	1	1	1	1	3
22	43	0	0	0	1	1	1	1	3
22	59	1	1	1	1	1	1	0	4
23	22	1	1	1	1	1	1	1	4
01	50	0	0	0	0	0	0	0	1

The preparation of the dataset was done by modeling the lighting habits of a home environment in which the students live. Once the dataset has been properly prepared, it has been used in training processes. The dataset using by the ML algorithms and successfully classifies the light conditions in case of the current system time. The output produced after this process is transmitted to the Arduino card via the serial port through the software. Thus, home lights can be successfully controlled via the AI module at specific time intervals when automation is required such as when there are no residents at home

Background

In this study, we used Scikit-learn which is one of the most popular frameworks used by python programming language for data acquisition and ML. It covers a wide range of algorithms for ML areas such as classification, clustering and regression.

The supervised learning method from among the six major learning methods widely used in ML was selected for use in this study.

A total of 5500 samples were generated in the home environment which were adapted to real life as well as to ML. The dataset used was 5500x10, n = 5500 samples, 9 features and a target. The dataset used here can express 9-input featured vectors and single output target vector in matrix form: $X \in R^{5500x9}$. Here, the X feature vector can be represented in a two-dimensional structure or in other words a 2D matrix.

$$X = \begin{bmatrix} x_{11} & \cdots & x_{19} \\ \vdots & \ddots & \vdots \\ x_{1n} & \cdots & x_{9n} \end{bmatrix} \quad (1)$$

Similarly, the target vector of the single-output model can express in matrix form:

$$y = \begin{bmatrix} y_{11} \\ \vdots \\ y_{1n} \end{bmatrix} \quad (2)$$

The main objective of the ML algorithms is to find the target function f and to match the input variables (X) to the output variables (y) using the best way: y=f(X). Various ML classification algorithms such as k-nearest neighbors (k-NN) classifier, Gaussian Naive Bayes (NB), Linear support vector machine (L-SVM), Decision tree classifier (DT), Linear discriminant analysis (LDA), and multi-layer Perceptron (MLP) algorithms were used in the study [12].

Since the problem here is a classification problem, classification algorithms are used for the solution. Therefore, some of the classification algorithms used in ML and preferred in this study are briefly explained below.

k-nearest neighbors (k-NN

The k-NN algorithm is a classification method in which the class with the sample data point and the nearest neighbors are determined according to k value [13, 14].

There, three different distance measurement methods were used in the k-NN algorithm for calculating the distance between two points. These were Euclidean distance, Manhattan distance and Chebyshev distance, respectively. In particular, Euclidean and Manhattan metrics were used extensively to measure the distance between predictor vectors [15].

The mathematical models of distance metrics used by the k-NN algorithm are as follows:

Euclidean distance between $x_i = [x_{i,1}... x_{i,n}]^T$ and $y_i = [y_{1,1}...y_{1,n}]^T$ is

$$d_{E(x_i,y_i)} = \sqrt{\sum_{i=1}^{n}(xi - yi)^2}. \quad (3)$$

Where x_i is ith feature independed variable and y_i is the target depended variable. The Manhattan distance between x_i and y_i is

$$d_{M(x_i,y_i)} = (\sum_{i=1}^{n}|x_i - y_i|^n)^{1/n}. \quad (4)$$

The Chebyshev distance is computed according to

$$d_{C(x_i,y_i)} = max_i(|x_i - y|) \quad (5)$$

Linear Discriminant Analysis (LDA)

In the study, instead of Logistic regression, which is a simple yet powerful classification algorithm, Linear discriminant analysis (LDA) algorithm, which gives advantage by size reduction in pre-processing operations, is preferred. In this model, since the linear classifier is composed of the weight vector expressed as w and bias expressed as b, the target vector y can be calculated as follows:

$$y = sign(w^T x + b) \quad (6)$$

Classification is carried out in two steps. First, the sample space is mapped to a one-dimensional space on the weight vector w. Then a dot is defined on the line to separate the two classes. Then LDA is used for the best values of w and b to separate the classes [16].

Naive Bayes (NB

A Naive Bayes classifier based on Bayesian theorem and composed of more than one algorithm can be used to train a dataset with large number of inputs. Despite being simple, its performance is quite satisfactory. Bayes' theorem can be expressed mathematically as follows:

$$P(y|X) = \frac{P(X|y)*P(y)}{P(X)} \quad (7)$$

Where, y is class variable and X is a dependent feature vector of our dataset (of size n=9). P(y|X) is the probability of hypothesis y given the data X. P(X|y) is the probability of data X given that the hypothesis was true. P(y) is the probability of hypothesis X being true (regardless of the data). P(X) is the probability of data (regardless of the hypothesis).

Support vector machine (SVM)

The most popular algorithm in modern machine learning, as a supervised machine learning algorithm, Support vector machine (SVM) can be used for both classification and regression problems. However, it is mostly used in classification problems. Here, the vectors that define the hyperplane are called support vectors. SVM does not perform well on very large datasets and computation costs are high. However, it performs very well on normal sized datasets [17]. The Standard SVM uses a linear decision boundary to classify the objects given by $w^y X_{new}$ + b. The SVM decision function for the X_{new} test point can be expressed as [18]:

$$y_{new} = sign(w^T X_{new} + b) \quad (8)$$

The LDA keeps the samples with the same class close but keeps the samples in the different classes away. For this process, on the one hand, the distance between the centers of different classes increases, while on the other hand the variance of each class decreases.

Multilayer perceptron (MLP)

One of the purposes of artificial intelligence is to make machines think like people who have extraordinary ability to learn new things. An

artificial neural network is a model based on the way the human brain works.

MLP is a fully connected and feed-forward artificial neural network (FFNN). The MLP consists of three layers: input, output and at least one hidden layer. Each layer which is composed of neurons and their connections, connected to adjacent layers. Intelligent neurons have the ability to calculate the sum of the weights of the inputs and give it to the final activation function [19].

Decisions Trees (DT)

As non-parametric, a supervised machine learning classification and regression algorithm, Decisions trees (DT) works in a split way and consists of a set of tree structure decision tests. The data in hand is continuously divided depending on a certain parameter. These algorithms usually operate as recursive processes. Since DT is simple to understand and to interpret, trees can be visualized. Given a training set X, the entropy of X is defined as [19]

$$Ent(X) = \sum_{y \in Y} P(y|X log P(y|X). \quad (9)$$

Results and discussion

All the classification algorithms used here have been tested using the Scikit-learn framework of the python programming language. In this study, a Windows 10 Pro based computer was used for all analyzes. The processor, memory and disk information of the computer are Intel i5 3.2 GHz, 8 GB RAM, 500 GB sata respectively. Accuracy of the proposed models can be calculated as,

$$Accuracy = \frac{N}{T} x100. \quad (10)$$

Here N and T are Number of lights correctly classified and Total number of lights respectively.

We will be confronted with a methodological error if the dataset used to train a model is also used for testing. This is called overfitting and may have a high success rate but it is not realistic. The main issue is to test the learning system with data that it has not yet come across and that is not within the scope of the training data. In general, a part of the dataset used for machine learning is used for training (i.e. 80%) and remaining is used for testing(i.e. 20%).

There is still an overfitting risk for datasets that are reserved for testing when different hyper parameters are applied to the system for the estimators. The system performance becomes doubtful when some parameters are changed manually since the system learns the test data. A different portion of the dataset must be retained as a validation set for overcoming this. However, the data to be used for training is decreased when the dataset is separated into three parts. A procedure called cross-validation (CV) is used for preventing this issue. The validation set is no longer needed when If CV is used. The dataset for training is divided into k small pieces called k-fold. The model is trained with k-1 training data and the remainder is used for validation.

When we take the k parameter for k-fold cross validation as n, the training dataset is divided into n parts for iteration of n. The n-1 fold is used for training while the remaining fold is used for testing purposes in order to evaluate system performance. The average estimated performance (e.g. classification accuracy or error) is calculated as follows.

$$P = \frac{1}{n} \sum_{i=1}^{n} P_i \quad (11)$$

The k-value has been accepted as 10 in the present application since the reasonable k-value in k-fold is taken as 10 for many applications.

Different distance metrics and different k values were used for the k-NN classifier algorithm in the Python environment for the dataset generated in the home environment by taking a k-fold value of 10. The best value for system performance in this case was 68.10% for Manhattan k = 25, 62.60% for Euclidean k = 16 and 45.20% for Chebyshev k = 13. As shown in Figure 2, the best overall performance values were obtained using the Manhattan metric. This is followed by Euclidean and Chebyshev metrics, respectively.

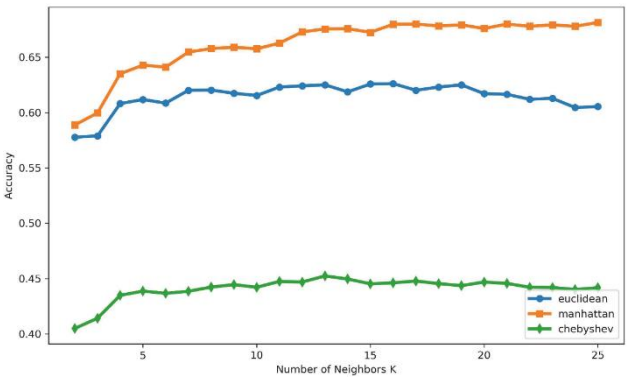

Figure 2. Comparation of k-NN metric accuracy vs k-value

Classification errors have been presented Figure 3 for the case where the k-NN algorithm was used with different metrics.

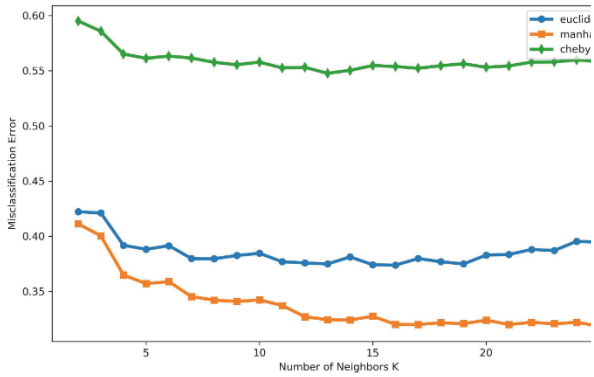

Figure 3. Comparation of k-NN metric error vs k-value

It appears that both the lowest success and the highest classification error for the values of 2 to 25 of k, the neighborhood value, occur when the Chebyshev metric is used. Besides, optimum error values for Euclidean, Manhattan and Chebyshev are obtained at k parameter values of 16, 25 and 13 respectively. The information in the case where more than one classification algorithm was used with different parameters has been shown in Table 2.

Table 2. Parameters of algorithms used for machine learning

Algorithms	Parameters	Values
MLP	activation	relu
	max_iter	600
	learning_rate	0.001
	hidden_layer_size	100
	solver	Sqd
L-SVM	kernel	linear
NB	Priors	None
LDA	Solver	Eigen
DT	max_depth	5
k-NN	n_neighbors	5

The performances for the six different classification algorithms used together have been given in Figure 4. The first three algorithms with the best performance are MLP with 96.69%, L-SVM with 94.98% and NB (91.23%). The lowest performance yields the k-NN algorithm with a value of 64.29%. Two other algorithms with moderate performance in classification are LDA and DT. Their performances were 87.54% and 74.18%, respectively. We believe that the dataset has a high number of records, which negatively affects the performance of the k-NN algorithm.

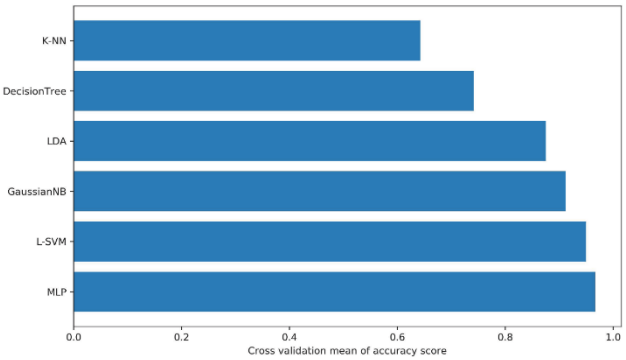

Figure 4. Comparison of machine learning classification algorithms using scikit-learn

Comparison of the other studies

Many studies have been carried out using classification algorithms. However, a comparison was made by taking into consideration various publications in which our algorithms were used. As was the case in [7], the best performance in this study was achieved with the MLP algorithm. As shown in Table 3, the classification performances obtained in our study are better in comparison with the results obtained from previous studies.

Table 3. Comparison of overall accuracy for classification

Reference paper	Dataset	Methodology and Classification Average
Cumin et all. [20]	Opportunity	MLP (90.21%) SVM (90.05%)
Vijayarani et all.[20]	synthetic kidney	NB (80.90%) SVM (80.0%)
Mazzoleni et all.[21]	Their own synthetic dataset	SVM (80.0%) DT (79.0%) NB (80.0%)
Takci et all. [22]	UCI statlog	SVM-L (84.81%) SVM-Sigmoid (84.41%)
This study	This synthetic dataset	MLP (96.69%) L-SVM (94.98) NB (91.23%) LDA (87.54%) DT (74.18%) NB (91.23%) k-NN (64.29%)

Conclusions and future work

In this study to begin with, intelligent home automation was established with Arduino-based

hardware and C# programming language-based software support. In the second place, a synthetic dataset, which was consisted of nine inputs, one output vector and about 5500 samples was created in accordance with real life. Moreover, system was trained with popular machine learning classification algorithms. To conclude, just like a home environment where real night life continued, the lights of the home was smartly managed to avoid thieves. Popular machine learning classification algorithms such as MLP, L-SVM, NB, DT k-NN were used to put forth the system success comparatively. In this study, MLP, L-SVM, and NB were similar but perform better than the others.

It is thought that homes protected by artificial intelligence will be safer, that the dataset quality will increase due to the data that will be acquired from a real home environment which in turn will yield more successful results.

Smart homes should be protected not only in the nights but also in the daytime. We would like to continue our work on this subject in future studies.

References

1. Benjamin K., Sovacool, Dylan D., Furszyfer DR. Smart home technologies in Europe: A critical review of concepts, benefits, risks and policies, Renewable and Sustainable Energy Reviews, Volume 120, 2020, pp.1-20.
2. Skn, H., Kalkan, H., Cetili, B. Classification of physical activities using accelerometer signals. In: Signal Processing and Communications Applications Conference, Mugla,Turkey,2012, pp.1-4.
3. Gne, H., Orta, E., Akda, D. Developing synthetic data generation software for artificial intelligence techniques used in smart home systems. Journal of Balikesir University Institute of Science and Technology, 18(2):1-11,2016.
4. Gariba, D., Pipaliya, B. Modelling human behaviour in smart home energy management systems via machine learning techniques. In: Proceddings of 2016 International automatic control conference,Taichung, Taiwan, 2016, pp. 53-58.
5. Dixit, A., Naik, A. Use of prediction algorithms in smart homes. International Journal of Machine Learning and Computing, 4:157-162, 2014.
6. Collado-Villaverde, A., R-Moreno, MD., Barrero, DF., Rodriguez, D. Machine learning approach to detect falls on elderly people using sound. In: Benferhat S., Tabia K., Ali M. (eds) Advances in Artificial Intelligence: From Theory to Practice. IEA/AIE. Lecture Notes in Computer Science, Springer, Cham, 2017, pp. 149-158.
7. Alshammari, T., Alshammari, N., Sedky, M., Howard, C. Evaluating machine learning techniques for activity classification in smart home environments. World Academy of Science, Engineering and Technology International Journal of Information and Communication Engineering, 12 (2):58-64, 2018.
8. Alhafidh, BAH., Allen, WH. Comparison and Performance Analysis of Machine Learning Algorithms for the Prediction of Human Actions in a Smart Home Environment. In:Proceedings of the International Conference on Compute and Data Analysis, New York, USA, 2017, pp. 54-59.
9. Dahmen, J., B. Thomas, DJ. Cook,X.Wang, Activity Learning as a Foundation for Security Monitoring in Smart Homes. Sensors ,17(4):pp. 1-17, 2017.
10. Cumin, J., Lefebvre, G., Ramparany, F., Crowley, FL. Human activity recognition using place-based decision fusion in smart homes. In:International and Interdisciplinary Conference on Modeling and Using Context, Paris, France, 2017, pp.137-150.
11. Singh, D., Merdiva, E., Hanke, S., Kropf, J., Geist, M., Holzinger, A. Convolutional and recurrent neural networks for activity recognition in smart environment. In:Towards integrative machine learning and knowledge extraction, Banff, AB, Canada, 2015, pp.194-205.
12. Jebakumari, VS., Shanthi, D., Sridevi, S., Meha, P. Performance evaluation of various classification algorithms for the diagnosis of Parkinson's disease. In:Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), Srivilliputhur, India, 2017, pp.1-7.
13. Cover, TM., Hart, PE. Nearest neighbor pattern classification. IEEE Transactions on Information Theory, IT13(1):2127, 1967.
14. Bhatia, N. Survey of nearest neighbor techniques. International Journal of Computer Science and Information Security, 8(2):302-305, 2010.
15. Mohana, TK., Lalitha, V., Kusuma, L., Rahul, N., Mohan, M. Various Distance Metric Methods for Query Based Image Retrieva. International Journal of Engineering Science and Computing, 7(3):5818-5821, 2017.
16. Zhou, ZH. Ensemble methods foundations and algorithms, Boca Raton, FL,2012.
17. Marshald, S. Machine learning an algorithmic perspectives, Boca Raton, FL, 2015.

18. Rogers, S., Girolami, M. A first course in machine learnings, Boca Raton, FL, 2012.
19. Sarkar, D., Bali, R., Sharma, T. Practical Machine Learning with Python, Bangalore, Karnataka, India, 2018.
20. Vijayarani, S., Dhayanand, S. Data mining classification algorithms for kidney disease prediction, International Journal on Cybernetics & Informatics (IJCI), 4(4):13-25, 2015.
21. Mazzolenia, M., Previdia, F., Bonfiglio, NS. Classification algorithms analysis for brain computer interface in drug craving therapy. Biomedical Signal Processing and Control 2017, https://doi.org/10.1016/j.bspc.2017.01.011.
22. Takci, H. Improvement of heart attack prediction by the feature selection methods, Turk J Elec Eng & Comp Sci,26:1-10, 2018.