Machine Learning Models for the Recognition of Commands in Smart Home Technologies[1]*

Lesia Mochurad[1,*,†], Viktoriia Babii[1,†], Michal Greguš[2,†]

[1] Lviv Polytechnic National University, 12 Bandera street, Lviv, 79013, Ukraine
[2] Faculty of Management, Comenius University Bratislava, Odbojárov 10, 820 05 Bratislava 25, Slovakia

Abstract
Researching the possibilities of using smart home technologies to support the independent life of lonely elderly people involves considering various aspects that can improve their comfortable and safe stay at home. For instance, health monitoring systems, access to medical information resources, nutritional care, home security, assistance in everyday affairs. The main purpose of this work is to research the application of data analysis methods and machine learning to develop a model that can recognize commands and perform various functions to ensure a comfortable and safe life for older people. Machine learning models used to recognize smart home commands such as Random Forest (RF), Support vector method (SVM), XGBoost, CatBoost, Multilayer Perceptron (MLP), Lightgradient boosted machine (LGBM) was analyzed. The models that were trained used data collected from Kaggle, a public repository of datasets. Optimal parameters in machine learning algorithms for the classification of smart home commands were determined. As a result of the comparative analysis conducted according to the experimental assessment, the MLP deep learning model demonstrated better results compared to other methods. The proposed model achieved accuracy at 91%. This indicator can be improved in the future by using different deep learning models.

Keywords
Smart home technologies, RF, SVM, XGBoost, CatBoost, MLP, LGBM

1. Introduction

The term 'smart home' is commonly used to refer to any environment equipped with a variety of technological systems and devices that allow you to automate and control various aspects of everyday life. Smart homes, equipped with sensors and actuators, serve to facilitate individuals in their everyday tasks, aiming to foster autonomy [1, 2]. Irrespective of disability status, these residences are tailored to accommodate anyone.

They incorporate technologies to monitor both the household environment and occupants, enabling communication between devices and providing assistance with daily routines [3]. This

SMARTINDUSTRY-2024: International Conference on Smart Automation & Robotics for Future Industry, April 18 - 20, 2024, Lviv, Ukraine
* Corresponding author.
† These authors contributed equally.
lesia.i.mochurad@lpnu.ua (L. Mochurad); viktoriia.babii.shi.2022@lpnu.ua (V. Babii); michal.gregus@fm.uniba.sk (M. Greguš)
0000-0002-4957-1512 (L. Mochurad); 0009-0008-6805-7915 (V. Babii); 0000-0001-6207-1347 (M. Greguš)

innovative approach holds significant potential in enhancing accessibility to home care services, especially for seniors and those with disabilities. 'Smart homes' provide the chance to assist elderly individuals in maintaining independence within their own residences as long as they can manage their own care needs.

Furthermore, smart homes offer assistance to elderly individuals with cognitive impairments, such as Alzheimer's disease or other forms of dementia, who struggle with daily tasks like eating, using the restroom, bathing, and dressing [5]. Typically, caregivers, family members, or professionals provide support and supervision to these individuals as needed. However, relying solely on caregivers can lead to feelings of frustration, anger, and helplessness among the elderly, particularly in sensitive situations like toileting [6].

Given that nursing homes have limited capacity and may not accommodate all individuals over the age of eighty, prolonging the duration of independent living at home holds both economic value and benefits for most individuals [7]. Consequently, a range of stakeholders, including governments, social service organizations, and even individuals and families, are increasingly embracing technological solutions to aid in the care of the expanding elderly population [8].

A systematic review detailed in [9] sought to investigate the utilization of smart home technologies in managing chronic diseases among the elderly. This study conducted a comprehensive literature review across four databases, following the PRISMA protocol, and examined nineteen articles. The identified intervention technologies were classified into three categories: smart home systems, external memory aids, and hybrid technologies.

The aim of this research is to determine the most effective method for recognizing and categorizing smart home commands using machine learning techniques. This study holds significance as it has the potential to improve the quality of life for elderly individuals and drive the development of smart home technology. The insights from this study could guide future advancements in smart home technology and contribute to enhancing the well-being of older adults.

2. Analysis of literary sources

The notion of social technology known as the "Smart house" has been evolving over several decades. In this context, smart homes represent a promising and cost-effective avenue for enhancing access to home care for the elderly and individuals with disabilities [10]. Numerous research groups have developed prototypes of smart homes or individual devices for integration into smart home environments. Primarily, these smart homes are tailored to cater to the needs of older individuals with various disabilities, encompassing motor, visual, auditory, or cognitive impairments [11].

One such development occurred in Finland, where a smart home utilizing capacitive internal positioning and contact sensors was created [12]. The goal was to recognize the activity of residents in a 69 m^2 smart home. The system they proposed used electrodes embedded in the floor to detect people at floor level and then analyze their interactions with household items such as tables, beds, refrigerators, and sofas.

Another notable example is the Great Northern Heaven Smart Home project in Ireland [13]. Sixteen smart homes were constructed to observe behavioral patterns in activities of daily living (ADL) among older adults. The system incorporated a combination of environmental sensor monitoring and manual data collection, alongside behavior recognition. Machine learning and pattern recognition techniques were utilized to analyze and forecast the well-being of residents.

In the Netherlands, researchers have developed the Aurama awareness system to facilitate aging in place, which includes various devices, including a photo frame, to monitor changes in behavioral patterns [14]. Additionally, technologies like radio frequency signals and bed sensors are employed to detect residents' presence and track their movements in and out of bed. Aurama has undergone three generations of development, each iteration refined based on feedback from field tests.

Researchers participating in the SPHERE project concentrate on sensing, networking, and machine learning applications for healthcare within hospital settings [15]. Their aim is to integrate sensor data to form comprehensive datasets for disease detection and treatment. SPHERE employs monitoring of physiological signals, home environment conditions, and vision-based surveillance. Their strategy combines a multimodal sensing system with intelligent data processing algorithms to manage data collection.

In the Netherlands, an Automatic Autonomous Surveillance (UAS) system has been developed to support aging in place [16]. This system relies on a ZigBee network and wireless sensors dispersed throughout the household. Additionally, cameras are utilized to trigger in emergency situations. The UAS has undergone testing by elderly individuals in Baarn and Soest, Netherlands.

Another initiative in the Netherlands involved setting up a test house [17]. Fourteen state-change sensors were installed throughout the house, and the testing period lasted 28 days, resulting in an annotated dataset accessible to the public.

In [18], a comprehensive description is provided for the foundational aspects of a smart home, including its architecture, utilization of essential devices, and integration of crucial technologies. The document extensively elaborates on the design of a smart home power supply system and associated communication systems.

In the paper [19], the authors conducted comprehensive research on the methods and tools used to detect cybersecurity in IoT devices using machine learning methods on various datasets.

The authors of the study [20] presented a model to improve the definition of activities in smart homes. The developed technique is based on the creation of a profile for each activity using training datasets. This profile served as the basis for the induction of additional functions that helped to distinguish the activity of residents, like fingerprints. To test the effectiveness of the proposed model, real datasets were used, and the results of the experiments showed a significant increase in accuracy compared to traditional methods.

In [21], the authors developed an algorithm for an autonomous robot that uses machine learning to open different types of doors with high accuracy. This algorithm can be used in smart homes to create robots that automatically open doors without the need for human assistance.

Our study is to determine the most effective approach for recognizing and classifying smart home commands using machine learning methods. The paper aims to improve the quality of life of the elderly and promote the development of smart home technologies.

Based on the performed analysis and comparison of the effectiveness of different machine learning models, the possibility of using new machine learning models will be investigated and the most effective model for recognizing smart home commands will be proposed, and its use for further development of smart home technologies will be recommended.

3. Methods and means

As widely acknowledged [22-25], machine learning techniques enable computers to "learn" from data and enhance their performance without explicit programming. These methods are employed to address a diverse range of tasks, spanning from classification and prediction to pattern recognition and automated analysis of data patterns.

3.1. Description and justification of selected machine learning models

Random Forest (RF) is a method that aggregates multiple decision trees trained on different subsets of the same complex dataset to mitigate variance. This comes at a slight increase in bias and some loss of interpretability but generally leads to significant improvements in model performance.

As is known, RF is a supervised learning algorithm that can be used for classification and regression tasks. It stands out for its flexibility and ease of use. The ensemble model comprises multiple decision trees, and the strength of the forest is said to increase with the number of trees it contains. RF constructs decision trees on randomly sampled data points makes predictions with each tree, and selects the most popular decision through voting. Additionally, it offers a reliable measure of feature importance.

Analyzing the algorithm, RF can be viewed as an ensemble technique that uses a divide-and-conquer approach to create decision trees on randomly partitioned data sets. This ensemble of classifiers forms a forest of decision trees. Each of the latter is constructed using attribute selection metrics such as information gain, gain ratio, or Gini index for each attribute. In regression problems, the average result of all trees serves as the final prediction, making RF simpler and more powerful than other nonlinear classification algorithms [26].

At its core, the RF leverages the wisdom of the crowd principle. In data science, this means that a multitude of relatively independent models (trees) acting collectively as a committee outperforms any individual model.

The dataset data is fed into the Random Forest classifier model, after which each tree from the forest makes a decision. Then the voting takes place and the model classifies (see Fig. 1).

Support vector machine (SVM) is a supervised machine learning algorithm that is capable of solving classification and regression problems, although it is mainly used for classification purposes. This high-performance technique operates by partitioning data into distinct regions.

In SVM classification, the algorithm constructs a line (or a plane/hyperplane in higher dimensions) within an N-dimensional space to separate data points belonging to two distinct classes. Points situated on one side of this line represent one class, while points on the other side pertain to the opposite class. The classifier endeavors to maximize the margin, which is the distance between the dividing line and the nearest data points of each class. This optimization aids in effectively delineating which points belong to which class.

Figure 1: Random forest functional model playback.

Support Vector Machine (SVM) represents another approach to binary classification problems and offers a range of kernel functions for flexibility [26]. The primary objective of the SVM model is to establish a hyperplane, or solution boundary, based on a set of features to classify data points. This hyperplane's dimension varies depending on the number of elements, and the task involves defining a plane in the N-dimensional space that maximally separates data points of two classes.

SVM aims to determine the maximum margin that effectively divides a dataset into two groups and predicts the category of new data points. Many individuals favor SVM due to its notable accuracy while requiring fewer computational resources. It excels particularly with smaller and more concise datasets and demonstrates efficiency in processing large-dimensional spaces while being memory-efficient.

XGBoost is a machine learning algorithm based on finding solutions using a decision tree and a gradient boosting framework. In unstructured data prediction tasks, artificial neural networks often outperform other algorithms or frameworks. However, for structured or tabular data of small size, algorithms based on decision tree search have a significant advantage. The infographic demonstrates the development of such algorithms.

XGBoost and Gradient Boosting Machines are ensemble methods based on boosting weak learners, usually binary decision tree algorithms, utilizing a gradient descent framework. Notably, XGBoost improves upon the GBM framework by optimizing the system and incorporating algorithmic enhancements.

In XGBoost, the construction of trees relies on parallelization, which is facilitated by the flexible nature of the cycles employed in building the foundation for learning. The outer cycle is responsible for listing the leaves of trees, while the inner cycle calculates the signs. However, nesting one cycle within another poses challenges for parallelizing the algorithm, as the outer cycle cannot commence execution until the inner one has completed its tasks. To enhance runtime efficiency, the order of cycles is adjusted: initialization takes place during data reading, followed by sorting using parallel threads. This reordering enhances algorithm performance by distributing computations across threads.

The algorithm is designed to make optimal use of hardware resources. This is achieved by creating internal buffers in each gradient statistics stream. Further improvements, for example, calculations outside the kernel, allow you to work with large datasets that are not contained in the computer's memory [27].

The Multilayer Perceptron (MLP) was developed to overcome this limitation. It functions as a neural network where the relationship between input and output is nonlinear. Unlike a perceptron, which typically relies on an activation function imposing a threshold like ReLU or sigmoid, MLP neurons have the flexibility to utilize any arbitrary activation function.

MLP consists of input and output layers, along with one or more hidden layers comprising numerous stacked neurons. It falls within the category of feedforward algorithms, similar to the perceptron, as input data combines with initial weights in a weighted sum and undergoes activation. However, the distinction lies in the fact that each linear combination extends to the subsequent layer, allowing for more complex transformations and nonlinear relationships to be learned.

Perceptrons create a single output based on several valid inputs, forming a linear combination using weights (sometimes passing the output through a nonlinear activation function). In terms of mathematics:

$$y = \varphi(\sum_{i=1}^{n} \omega_i x_i + b) = \varphi(w^T \mathrm{x} + b),$$

where w denotes a weight vector, x — a vector of input data, b — displacement, φ — a nonlinear activation function.

Catboost — a machine learning method based on gradient boosting. Almost any modern method based on gradient boosting works with numerical signs. If in our dataset there are not only numerical, but also categorical features, then it is necessary to translate the categorical features into numerical ones. This leads to a search for their essence and a potential decrease in the accuracy of the model. That is why it was important to develop an algorithm that should work not only with numerical features, but also with categorical directions, the patterns between which this algorithm will be independently detected.

Categorical attributes consist of distinct values, referred to as categories, which are not inherently comparable, rendering them unsuitable for direct use in binary decision trees. A common approach with categorical attributes involves converting them into numerical values during preprocessing, where each category in every example is replaced by one or more numerical representations.

The prevalent technique, typically applied to categorical features with a limited number of categories, is one-hot encoding: the original attribute is removed, and a new binary variable is introduced for each category. Simultaneous encoding can occur during preprocessing or training; the latter can be more efficient in terms of training time and is implemented in CatBoost [28].

CatBoost employs a more efficient strategy that minimizes conversion and enables the utilization of the entire dataset for training. Specifically, it involves randomly permuting the dataset and calculating, for each example, the average label value of examples with the same category value placed earlier in the permutation. Employing multiple permutations can further enhance effectiveness. However, directly utilizing statistical data from multiple permutations may lead to overfitting. As discussed in the subsequent section, CatBoost employs a novel approach to compute leaf values, which allows for the incorporation of multiple permutations without encountering this problem.

LightGBM is an ensemble method employing a tree-like learning algorithm. In contrast to other tree-based learning algorithms that expand horizontally (by levels), LightGBM grows trees vertically, adding leaves sequentially.

Using a tree-like algorithm to grow a tree will result in the same final tree result. The difference here lies in the order from which the tree is developed and the stopping measures such as the number of trees to create and the reduction methods that would change the final result of the decision tree.

LightGBM allows you to use more than 100 hyperparameters that can be customized to your liking.

LightGBM is a gradient-based improvement framework that uses tree-based learning algorithms. It has the following advantages:

- Shorter training time and higher efficiency.
- Less memory usage.
- Better accuracy.
- Support for parallel and distributed learning.
- Ability to handle big data.

Comparative experiments conducted on publicly available datasets demonstrate that LightGBM exhibits superior efficiency and accuracy compared to existing frameworks while consuming significantly less memory. Additionally, distributed learning experiments reveal that LightGBM can achieve linear acceleration by leveraging multiple learning machines in specific configurations.

3.2. Indicators of model performance evaluation

Assessing the performance of a machine learning model is a crucial aspect of model development. Therefore, it is essential to understand how the success of a machine-learning model can be gauged.

Evaluation metrics are tailored to specific machine learning tasks, with various metrics available for classification problems. Employing diverse metrics to evaluate performance enables the comprehensive assessment of a model's efficacy before deploying it for real-world data processing.

Classification tasks involve predicting class labels based on input data, with binary classification entailing only two possible output classes. There exists a multitude of methods for measuring classification performance, including popular metrics such as accuracy, confusion matrix, and AUC-ROC. Additionally, precision and recall are commonly utilized metrics in classification problems.

Accuracy measures how often a classifier predicts correctly. We can define accuracy as the ratio of the number of correct predictions to the total number of predictions. Accuracy gives us an overall picture of how much one can rely on model predictions. However, a balanced dataset is required to use this metric.

The confusion matrix allows you to tabulate the number of correct and incorrect predictions made by the model compared to the actual classifications in the test set, showing the types of errors that occur. This matrix evaluates the model's effectiveness in classifying test data with known true values, typically in an *n x n* format, where *n* represents the number of classes. It is constructed after the test data has been predicted.

There are four possible outcomes of classification prediction:

- True positive outcomes (TP). These are actual positive results that are predicted to be positive.
- False negatives (FN). These are actual negative results that are predicted to be negative.
- False positives (FP). These are actual negative results that were predicted to be positive (type one errors).
- False negatives (FN). These are actual positive results that were predicted to be negative (type two errors).

Precision quantifies the proportion of correctly predicted positive cases out of all predicted positives. Accuracy is particularly relevant when false positives are of greater concern than false negatives, calculated as the ratio of true positives to predicted positives.

Recall measures the proportion of actual positive cases correctly predicted by the model. It is valuable when false negatives are more critical than false positives and is calculated as the ratio of true positives to the total number of actual positives.

The F1 score represents the harmonic mean of Precision and Recall, peaking when Precision equals Recall.

4. Numerical experiments

4.1. Description of the dataset

For this study, a dataset [30] was used, which contains information about the ‘smart home’ commands and the corresponding attributes of this command. The dataset was taken from the Internet resource Kaggle. Total number of records - 6663. After preliminary analysis of the dataset, attributes that are not necessary for further data processing were removed. The input column will be a Sentence column, which means a person's request for a smart home. The target variables are Category, Subcategory, and Action columns. That is the model returns the output values of the category, subcategories and actions for the incoming user request.

Dataset structure: The Category column contains the name of the category to which the command belongs. For instance, it can be ‘lighting’, ‘climate control’ or ‘security’. The Action_needed column contains a binary value (1 or 0) that indicates whether additional user action is required after the command is executed. If the value is 1, then the user needs to perform an additional action (for example, press the button to turn on the lighting). If the value is 0, no additional action is required. The Question column contains a binary value (1 or 0) indicating whether the command contains a question. If the value is 1, then the command contains a question (such as ‘What is the temperature in the room?’). If the value is 0, the command does not contain a question. The Subcategory column contains the command subcategory (if any). For example, in the category ‘lighting’ there may be subcategories ‘lamp’, ‘chandelier’ or ‘night traffic light’. The Action column contains a description of the action to be performed (such as ‘turn on’, ‘turn off’ or ‘increase the temperature’). The Time column contains the time when the command was sent. The Sentence column contains a textual description of the command.

4.2. Results of data analysis and preliminary processing

Having made a detailed review of the data of the selected dataset, it was found that raw data may not give effective accuracy, and may also complicate their understanding by the model. Therefore, the next step is data preprocessing. This approach helps to remove everything unimportant from the dataset and prepare the data for further processing. The representation of the decoded word can be performed simply by initializing the vector with all zeros and placing one instead of that word in the dictionary.

The distributed representation of words is a little more complicated, the neural network helps with this. Word2vec is used to obtain these distributed word embeddings. Word2vec uses skip-gram model. It takes the weight vector between the input layer and the hidden layer after learning each word with its nearest neighbors. Taking the weight matrices of this neural network, the hidden representations of the words were encapsulated in the vector representations of each word. These hidden representations between words are already embedded in the word vector.

To obtain a training vector for learning machine learning methods, the sentence must be converted into a vector. Each sentence consists of several words, each of which has its own word. Word representations can be combined into sentence representations, this can be done in several ways: by simply averaging word vectors or first multiplying by a TF-IDF score (term frequency, inverse of document frequency), and then averaging. This score is obtained by multiplying the frequency of the term by the inverse frequency of the document. The term-frequency is the probability of a word appearing in a sentence, and the inverse of the document frequency is used to indicate how rare a word is in a sentence. This avoids giving more importance to sentences where the same word occurs several times.

In machine learning, numerical data is often preferred, necessitating the conversion of text data into numerical vectors through a process called vectorization, especially in natural language processing tasks.

TF-IDF vectorization involves computing the TF-IDF score for each word in a corpus relative to a specific message and then constructing a vector based on this information. Consequently, each message in the corpus is represented by its own vector, where each element corresponds to the TF-IDF score of a word across the entire collection of messages. These vectors find utility in various applications; for instance, one can assess the similarity between two documents by comparing their TF-IDF vectors using cosine similarity.

Stages of data vectorization:

- creating a dictionary containing words from the entire dataset;
- calculates how often a word occurs in a sentence;
- calculates how often a word appears in the entire dictionary;
- the value of the frequency term opposite to the TF-IDF document is obtained;
- obtaining encoded vectors for the target variable;
- obtaining encoded vectors for words;
- the encoded integer sentence vector is finalized.

TF (term frequency) measures the proportion of specific words appearing in a document relative to the total word count in that document. IDF (inverse document frequency) represents the inverse of the frequency of a particular word across all documents in a corpus.

In Fig. 2, two columns are presented. The first column contains pairs of numbers: the index of the sample element and the unique token linked to that element. The numbers in the second column denote the computed TF-IDF values, reflecting the importance of each word within the text.

```
(0, 11377)    0.18021639269164577
(0, 16434)    0.6235621903502868
(0, 5111)     0.354164663804394
(0, 12318)    0.423894334925617
(0, 5150)     0.41393942834232855
(0, 3369)     0.31973016742978555
(1, 14407)    0.43105698238566925
(1, 697)      0.3381977708126418
(1, 3341)     0.33736027286851644
(1, 10848)    0.382705381258259
(1, 337)      0.39738827482301653
(1, 7450)     0.2937747098704154
(1, 12985)    0.1647232877217583
(1, 12968)    0.2841352179864129
(1, 11377)    0.11570068980479913
```

Figure 2: TD-IDF statistic value for each word.

The next stage is the use of the train_test_split function, which is designed to separate one dataset for two different purposes: learning and testing. The testing subset is designed to build a model. A subset of testing is designed to use a model on unknown data to estimate model performance. Consequently, the dataset was divided in a proportion of 80/20 respectively.

4.3. Evaluation the efficiency of selected models

The results of the classification of smart home commands by machine learning methods will be compared next, namely RF, SVM, XGBoost, MLP, LGBM and CatBoost. To compare machine learning models, scoring metrics such as accuracy, precision, recall, and f-score are used. A comparison of the accuracy of machine learning models is shown in Table 1.

Table 1
Results of machine learning classifiers work

Classifier	Precision	Recall	F1-score	Accuracy
RF	0.816	0.813	0.814	0.879
SVM	0.830	0.855	0.842	0.867
XGBoost	0.953	0.938	0.945	0.903
CatBoost	0.865	0.842	0.853	0.879
MLP	0.911	0.912	0.903	0.909
LGBM	0.756	0.723	0.753	0.716

As can be seen from Table 1, RF has good results with accuracy, completeness and F1-indicator, as well as high accuracy. SVM also has good results with accuracy, completeness and F1-indicator. It achieves high accuracy but slightly lower than RF.

SVM nevertheless shows high completeness, which means that it recognizes positive classes well. XGBoost shows good overall results, making it a strong option for classifying smart home commands. CatBoost also has quite well results with accuracy, completeness and F1-indicator, but its accuracy is lower compared to previous models. It is also worth noting that it has high accuracy (Accuracy), which means that it predicts classes well in general.

MLP has very good results with accuracy, completeness and F1-indicator. It achieves the highest accuracy of any model under consideration, making it a potentially strong choice for smart home command classification. LightGBM has the lowest results among all models with accuracy, completeness and F1-indicator.

5. Discussion of study results

The MLP model, a neural network type, is employed to address classification, regression, and various other machine learning tasks. In this study's context, the MLP was utilized to categorize data collected from sensors within an elderly person's residence. The assessment of this model's outcomes involves scrutinizing its accuracy and effectiveness. Typically, diverse metrics such as accuracy, sensitivity, specificity, and the F1-score are employed to evaluate model performance.

For this particular problem, the accuracy of the MLP model can be gauged by comparing the sensor data from the household with the model's predicted data. For instance, if the model accurately predicts the presence of an elderly person when they are indeed at home, it's considered a successful prediction. Conversely, if the model incorrectly assumes a person's presence in the house when they are not, it may indicate the necessity for additional model refinement.

To improve the operation of the MLP model, various techniques can be used, such as increasing the number of layers and neurons in the network, using various optimization algorithms, changing learning parameters, using data augmentation and others.

Further experiments were conducted to investigate the effectiveness of MLP for classifying smart home commands using different target variables. Three different target variables were used in three cases. The results are presented in Table 2.

There should be no empty lines before section headings. The template already adds the necessary spacing before them.

Table 2
MLP model accuracy table on test data

MLP	Category	Subcategory	Actions
Accuracy	0.909	0.868	0.904

It must be noted that a model with a category as a target variable and a model with an action as a target variable have almost the same accuracy, which may indicate a similar importance of these variables in classification.

A model with a subcategory as a target variable has the lowest accuracy among the three, which may indicate the difficulty of classifying subcategories compared to categories or actions.

In general, MLP shows good results for the classification of smart home commands, regardless of the target variable, but category and action may be the most important factors for classification with high accuracy.

To assess the effectiveness of models with different target variables, an error matrix was used as an estimation metric. In Fig. 3 the confusion matrix for the MLPClassifer model that was trained using the category as the target variable is shown.

The confusion matrix reflects the error values between true classes and predicted classes. The map has a dimension corresponding to the number of unique classes. Within each entry of the matrix, the number of examples is displayed, where the real class and the predicted class coincide. Error values (number of incorrect predictions) are represented by numbers in each entry.

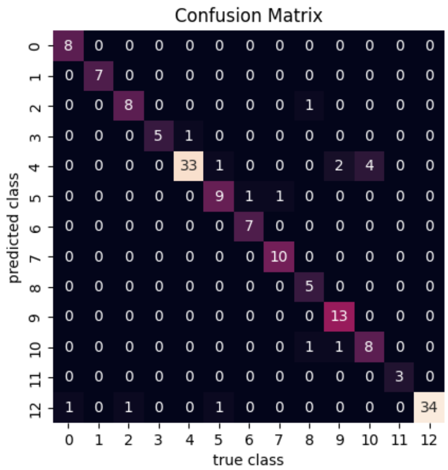

Figure 3: Confusion matrix for categories.

Diagonal (see Fig. 3) contains the largest values, which means correctly classified objects. In Fig. 3 it can be seen that four values belonging to the fourth class were incorrectly classified by the model as values from the tenth class.

Figure 4 displays the confusion matrix for the MLPClassifier model, which was trained using the subcategory as the target variable. The test data consists of 17 classes. Each entry in the matrix represents the number of examples where the actual class and the predicted class are the same. Entries beyond the diagonal of the matrix indicate the number of examples where the actual class and the predicted class differ.

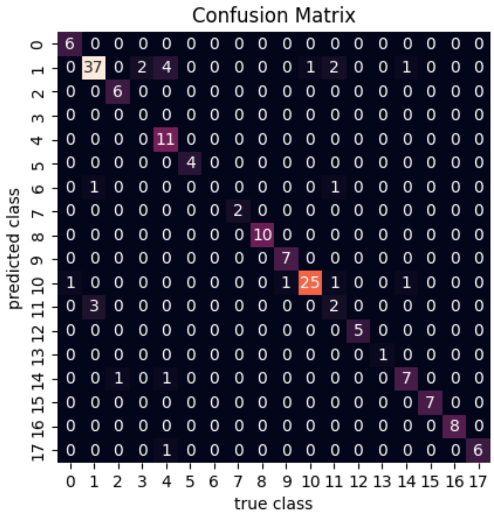

Figure 4: Confusion matrix for subcategories.

Similarly, in Fig. 5 is a confusion matrix tested on data with an action target variable.

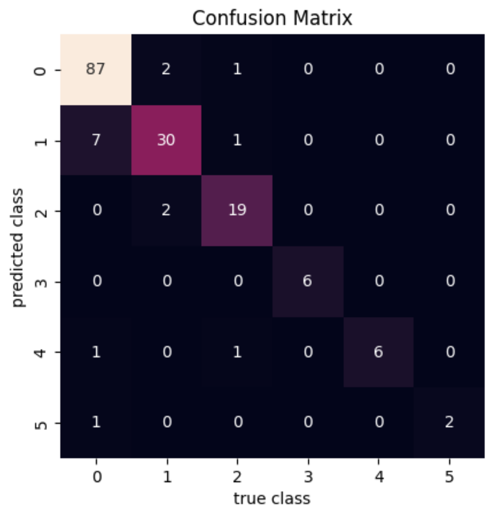

Figure 5: Confusion matrix for actions.

Test experiments were conducted using MLP models. The prediction process is based on using the trained MLPClassifer model to classify the incoming sentence into a category, subcategory, and action.

In Fig. 6 the prediction results of MLPClassifer models for various sentences that contain smart home commands are shown. The original sentence is displayed first. Then a category for this sentence is provided. Below is the predicted subcategory for this sentence and the predicted action for this sentence.

```
Sentence ->  Hi Google, please turn off the lights.

Predicted category ->  lights
Predicted subcategory ->  none
Predicted action ->  off

Sentence ->  Turn the lights off in the kitchen.

Predicted category ->  lights
Predicted subcategory ->  kitchen
Predicted action ->  off

Sentence ->  Random sentence.

Predicted category ->  other
Predicted subcategory ->  none
Predicted action ->  none

Sentence ->  Lower the door.

Predicted category ->  shutters
Predicted subcategory ->  garage
Predicted action ->  down
```

Figure 6: MLP model prediction results.

6. Summary and Conclusion

The implementation of smart homes can lead to numerous benefits in society, in particular ensuring continuity of care and continuous monitoring of the elderly by local and hospital health services. In addition, ‘smart homes’ are aimed at providing convenience and facilitating everyday life for a wide range of people. The study of methods of algorithms turns out to be a key stage in the development of such intelligent homes. The smart home system should be able to analyze user actions and behavior to identify patterns and use them to ‘predict’ future user behavior.

In this study the most common analysis methods used in smart home technology over the past five years are considered. At the initial stage, popular machine learning algorithms were identified and used, in RF, SVM, XGBoost, MLP, LGBM and CatBoost. The data these models were trained on came from Kaggle, a publicly available dataset. Optimal parameters for machine learning algorithms were determined in order to classify commands for a smart home.

Experimental evaluation demonstrated that the MLP deep learning model performed better compared to individual methods. Developed model achieved 91% accuracy. This accuracy can be further improved using various deep learning models. Additional tuning of machine learning classifier hyperparameters may lead to even better results in the future.

Acknowledgements

The authors express their gratitude to the Armed Forces of Ukraine for ensuring the security necessary to carry out this work. The accomplishment of this work was made possible solely due to the determination and bravery exhibited by the Ukrainian Army.

References

[1] J. Liao, X. Cui, H. Kim. Mapping a Decade of Smart Homes for the Elderly in Web of Science: A Scientometric Review in CiteSpace. Buildings. 2023; 13(7):1581. https://doi.org/10.3390/buildings13071581.

[2] V.V. Shkarupylo, I.V. Blinov, A.A., Chemeris, *et. al.* On Applicability of Model Checking Technique in Power Systems and Electric Power Industry. Studies in Systems, Decision and Control, book series. 2021; 399:3–21.

[3] R. Turjamaa, A. Pehkonen, & M. Kangasniemi, How smart homes are used to support older people: An integrative review. International journal of older people nursing, 2019, 14(4), e12260. https://doi.org/10.1111/opn.12260.

[4] S.-A. Precup et al., Recognising Worker Intentions by Assembly Step Prediction, 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA), Sinaia, Romania, 2023, pp. 1-8, doi: 10.1109/ETFA54631.2023.10275423.

[5] P. Tiwari, V. Garg, R. Agrawal, Changing World: Smart Homes Review and Future. In: Moh, M., Sharma, K.P., Agrawal, R., Garcia Diaz, V. (eds) Smart IoT for Research and Industry. EAI/Springer Innovations in Communication and Computing. Springer, Cham. 2022, https://doi.org/10.1007/978-3-030-71485-7_9.

[6] Sanchez V.G., Pfeiffer C.F., Skeie N.-O. A Review of Smart House Analysis Methods for Assisting Older People Living Alone. Journal of Sensor and Actuator Networks. 2017; 6(3):11. https://doi.org/10.3390/jsan6030011.

[7] Randerath, J. (2023). Syndromes of limb apraxia: Developmental and acquired disorders of skilled movements. In G. G. Brown, T. Z. King, K. Y. Haaland, & B. Crosson (Eds.), APA handbook of neuropsychology, Vol. 1. Neurobehavioral disorders and conditions: Accepted science and open questions (pp. 159–184). American Psychological Association. https://doi.org/10.1037/0000307-008.

[8] S. Rossi, A. Coppola, M. Gaita and A. Rossi, "Human–Robot Interaction Video Sequencing Task (HRIVST) for Robot's Behavior Legibility," in IEEE Transactions on Human-Machine Systems, vol. 53, no. 6, pp. 975-984, Dec. 2023, doi: 10.1109/THMS.2023.3327132.

[9] Facchinetti, G., Petrucci, G., Albanesi, B., *et. al.* (2023). Can Smart Home Technologies Help Older Adults Manage Their Chronic Condition? A Systematic Literature Review. International journal of environmental research and public health, 20(2), 1205. https://doi.org/10.3390/ijerph20021205.

[10] Amiribesheli, M.; Benmansour, A.; Bouchachia, A. A review of smart homes in healthcare. J. Ambient Intell. Humaniz. Comput. 2015, 6, 495–517.

[11] Chan, M.; Estève, D.; Escriba, C.; Campo, E. A review of smart homes—Present state and future challenges. Comput. Methods Programs Biomed. 2008, 91, 55–81

[12] Valtonen, M.; Vuorela, T.; Kaila, L.; Vanhala, J. Capacitive indoor positioning and contact sensing for activity recognition in smart homes. J. Ambient Intell. Smart Environ. 2012, 4, 305–334.

[13] Doyle, J.; Kealy, A.; Loane, J.; *et. al.* An integrated home-based self-management system to support the wellbeing of older adults. J. Ambient Intell. Smart Environ. 2014, 6, 359–383.

[14] Dadlani, P.; Markopoulos, P.; Sinitsyn, A.; Aarts, E. Supporting peace of mind and independent living with the Aurama awareness system. J. Ambient Intell. Smart Environ. 2011, 3, 37–50.
[15] Zhu, N.; Diethe, T.; Camplani, M.; *et. al*. Bridging e-health and the internet of things: The sphere project. IEEE Intell. Syst. 2015, 30, 39–46.
[16] Van Hoof, J.; Kort, H.; Rutten, P.; Duijnstee, M. Ageing-in-place with the use of ambient intelligence technology: Perspectives of older users. Int. J. Med. Inf. 2011, 80, 310–331.
[17] Van Kasteren, T.; Noulas, A.; Englebienne, G.; Kröse, B. Accurate activity recognition in a home setting. In Proceedings of the 10th international conference on Ubiquitous computing, Seoul, Korea, 21–24 September 2008; pp. 1–9.
[18] Min Li, Wenbin Gu, Wei Chen, *et. al*. Smart Home: Architecture, Technologies and Systems, Procedia Computer Science, Volume 131, 2018, Pages 393-400, https://doi.org/10.1016/j.procs.2018.04.219.
[19] Hulayyil SB, Li S, Xu L. Machine-Learning-Based Vulnerability Detection and Classification in Internet of Things Device Security. Electronics. 2023; 12(18):3927. https://doi.org/10.3390/electronics12183927.
[20] Majdi Rawashdeh, Mohammed G.H. Al Zamil, Samer Samarah, M. Shamim Hossain, Ghulam Muhammad, A knowledge-driven approach for activity recognition in smart homes based on activity profiling, Future Generation Computer Systems, Volume 107, 2020, Pages 924-941, https://doi.org/10.1016/j.future.2017.10.031.
[21] L. Mochurad, Y. Hladun, Y. Zasoba, M. Gregus. An Approach for Opening Doors with a Mobile Robot Using Machine Learning Methods. Big Data Cogn. Comput. 2023, 7, 69. https://doi.org/10.3390/bdcc7020069.
[22] D. Chumachenko, M. Butkevych, D. Lode, M. Frohme, K. J. G. Schmailzl, and A. Nechyporenko, “Machine Learning Methods in Predicting Patients with Suspected Myocardial Infarction Based on Short-Time HRV Data,” Sensors, vol. 22, no. 18, p. 7033, Sep. 2022, doi: https://doi.org/10.3390/s22187033.
[23] M. Mazorchuck, V. Dobriak, and D. Chumachenko, “Web-Application Development for Tasks of Prediction in Medical Domain,” 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), pp. 5–8, Sep. 2018, doi: https://doi.org/10.1109/stc-csit.2018.8526684.
[24] . Lesia Mochurad, Rostyslav Panto. A Parallel Algorithm for the Detection of Eye Disease. CSDEIS 2022, LNDECT 158, pp. 1–15, 2023. https://doi.org/10.1007/978-3-031-24475-9_10.
[25] L. Mochurad, A. Solomiia. Optimizing the Computational Modeling of Modern Electronic Optical Systems. Lecture Notes in Computational Intelligence and Decision Making. ISDMCI 2019. Advances in Intelligent Systems and Computing, 2020, vol 1020. Springer, Cham. pp 597-608. doi: 10.1007/978-3-030-26474-1_41.
[26] D. P. Mohandoss, Y. Shi and K. Suo, "Outlier Prediction Using Random Forest Classifier," 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), NV, USA, 2021, pp. 0027-0033, doi: 10.1109/CCWC51732.2021.9376077.
[27] Atin Roy, Subrata Chakraborty, Support vector machine in structural reliability analysis: A review, Reliability Engineering & System Safety, Volume 233, 2023, 109126, https://doi.org/10.1016/j.ress.2023.109126.

[28] Yixiao Zhang, Zhongguo Zhao, Jianghua Zheng, CatBoost: A new approach for estimating daily reference crop evapotranspiration in arid and semi-arid regions of Northern China, Journal of Hydrology, Volume 588, 2020, https://doi.org/10.1016/j.jhydrol.2020.125087.

[29] M. R. Machado, S. Karray and I. T. de Sousa, "LightGBM: an Effective Decision Tree Gradient Boosting Method to Predict Customer Loyalty in the Finance Industry," 2019 14th International Conference on Computer Science & Education (ICCSE), Toronto, ON, Canada, 2019, pp. 1111-1116, doi: 10.1109/ICCSE.2019.8845529.

[30] Smart Home Commands Dataset. Available Online: https://www.kaggle.com/datasets/bouweceunen/smart-home-commands-dataset (accessed on 19 December 2023).