Controlling Home Appliances Adopting Chatbot Using Machine Learning Approach

Mojammel Hossen, Minhazul Arefin, Rakib Hossen and Mohammed Nasir Uddin

March 9, 2022

Controlling Home Appliances adopting Chatbot using Machine Learning Approach

Kazi Mojammel Hossen, Minhazul Arefin, Rakib Hossen and Mohammed Nasir Uddin

Abstract In the last decades, home automation becomes popular and rapidly increased artificial intelligence-based controlling systems. So, many researchers have been interested in the Internet of things so that every appliance should be autonomous. Smart home technology is one of them. It involves certain electrical and electronic systems in a building with some degree of computerized or automated control. It can control elements of our home environments (e.g. light, fans, electrical devices, and safety systems). We propose an approach that fully controlled the home appliances by chatbot technology. In our research, the system can extract the device name such as light, fan, etc using synonyms. In the device name extraction part, we use Jaro-Winkler string matching algorithms. We have also used the Naive Bayes algorithm to take command for action. Finally, a Firebase-based system connects the users and controls hardware. Our model can control the home appliances from a long distance because we used the wireless fidelity system.

Key words: Home Automation . Internet of Things . Machine Learning . Chatbot . Natural Language Processing

Kazi Mojammel Hossen
Jagannath University, Dhaka, Bangladesh e-mail: `mojammel.jisan@gmail.com`

Minhazul Arefin*
Jagannath University, Dhaka, Bangladesh e-mail: `minhazularefin21@gmail.com`

Rakib Hossen
Jagannath University, Dhaka, Bangladesh e-mail: `rakibcse5003@gmail.com`

Mohammed Nasir Uddin
Jagannath University, Dhaka, Bangladesh e-mail: `nasir@cse.jnu.ac.bd`

1 Introduction

Automation is a strategy, method, or system for using electronic equipment to operate or control a process while minimizing human participation. Home automation is a new concept that brings together science and engineering from several areas [1]. Home automation is not a new term for science society. Research has been going on for more than a decade. In this paper, We concentrate on Intelligent Conversational Software Agents or Chatbots [2, 3] with a machine learning approach. The model includes four parts: firstly, the Chatbot technology used to works on Natural Language Processing (NLP). Secondly, a mobile application can conduct a conversation with a user via text. Next, in the device name extraction part, we use Jaro-Winkler string matching algorithms [4]. In a high-level language, it informs the user about device status and takes command for action where Chatbot classifies texts using the Naive Bayes algorithm [5] Finally, a Firebase-based system connects the users and controls hardware. The main goal of this research is to design a system that convert Natural Language commands into hardware commands to operate the IoT device in a house.
A natural language-based (English) home automation system that will work through the internet and also enable one to remotely manage his/her appliances from anywhere, anytime. On the other hand, the device name retrieved from the text should work as intended, even if users use synonyms such as light, which has several synonyms. We also want to enhance security for IoT device control from a long distance. The demand for mobile applications is continuously increasing because of the continuous evolution of the mobile device's popularity and functionality. Using several forms of connection mechanisms, the smartphone application is utilized to manage and monitor home appliances. It must offer the devices facility to talk to each other. Indeed, users are using mobile applications on their phones to control their houses from a distance. This system is useful to help handicapped and aged people that control home appliances.

2 Background Study

ELIZA is widely regarded as one of the first programs to pass the Turing test. ELIZA began by examining the text entered by the user and searching for specific keywords. The input was then turned into a response by applying values to them. Richard Wallace's ALICE (Artificial Linguistic Internet Computer Entity) is an open-source natural language processing chatbot application that evaluates user input using certain heuristical patterns matching rules and converses with a human. Siri [6] converts human voice to text using ASR (Automatic Speech Recognition). It scans parsed text and detects user commands and activities using question and purpose analysis. Like schedule a meeting or set my alarm. Alexa [7] is a voice service that comes pre-installed on the Amazon Echo gadget. Alexa uses natural language processing technologies for voice interaction.

3 Related Works

E. Kasthuri introduced a long short term memory method for natural language processing and deep learning chatbots at el. [8]. A chatbot for the lab handbook has been added to the recommended system [9]. Students will be unable to pick the best choice because there are so many possibilities. Questions can be made, and Google can supply answers; nevertheless, since there are so many options, students will be unable to choose the best option. Questions can be answered in a multitude of ways for one response, but when our goal is limited down, the answer will be one. Based on their ability to train a model to properly answer a student's query, teachers can predict the chance of a question being presented for one inquiry. To get a specific answer to a question posed by a student during a practical. To get a specific answer to a question posed by a student during a practical. Because most sessions are now done on an internet platform, students may meet issues during the practical session, such as how to use this program, how to install the software, and so on. A certain teacher can generate potential questions and replies to those questions. In the recommended system, an interactive education-oriented chatbot [10, 11] is built.
A chatbot service system based on fuzzy logic principles and fuzzy inference was proposed by Nicholas at el. [12]. It was built for the Covenant University Doctor (CUDoctor) telemedicine system. In Nigeria, the service assesses the symptoms of tropical illnesses [13]. The chatbot and the system were connected using the Telegram Bot Application Programming Interface (API), while the system and a short message service (SMS) subscriber were connected using the Twilio API. The service makes use of a knowledge base derived from medical ontologies that contains known information about illnesses and symptoms. The illness is efficiently predicted using a fuzzy support vector machine (SVM) based on the symptoms supplied. NLP recognizes the users' inputs and forwards them to the CUDoctor for decision support [14]. Finally, the user receives a message indicating the completion of the diagnosis procedure.
R. Pradeep at el. [15] presented a chatbot that focuses on developing an AI-based automated system called Medbot, which leverages NLP and Machine Learning to generate a customised Virtual assistant for answering questions regarding medical devices. It assumes the role of technical support personnel in terms of comprehending the specific operations and features of medical equipment, which is sometimes more difficult to manage. The Medbot answers the user's question quicker than a traditional system that skims through an entire manufacturer's documentation [16].
Vipul at el. [17] provide a simple way for business users to comprehend that may be readily implemented for linking the chatbot with a business intelligence platform This communication explains how to use artificial intelligence technology, namely NLP, to have a conversation with a chatbot [18]. The introduction of the chatbot and its connection with the BI tool [19] has shown promising results.

4 IoT System Architecture

The architecture of an Internet of Things system is a multi-stage process in which data is delivered from sensors connected to ”things” to a network, then processed, analyzed, and stored in a corporate data center or the cloud. A ”thing” in the Internet of Things might be a machine, a structure, or even a person. In order to govern a physical process, processes in the IoT architecture also transfer data in the form of instructions or commands, which inform an actuator or other physically linked device how to accomplish a task. An actuator can perform basic tasks such as turning on a light or more sophisticated tasks such as shutting down a production line if a problem is discovered.
The chatbot algorithm’s architectural framework is stored on a web server. The Chatbot’s application will be connected to the webserver. The requests will be received by the client application, which will process them on the server before delivering them to the hardware (Raspberry Pi 3) for control. The hardware (Raspberry Pi 3) sends an acknowledgment signal to the server and, as a result, to the client once the job is done. As a result, the user gets told when the task is completed successfully. Users can directly communicate with the chatbot by giving it instructions and receiving responses. Once the task is completed, the hardware(Raspberry Pi 3)

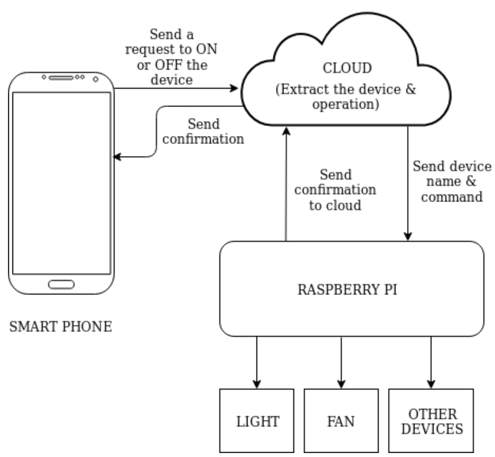

Fig. 1: System Architecture

sends an acknowledgment signal back to the server and hence to the client. As a result, the user gets notified that the task is completed successfully. Users can talk to the chatbot directly by doing input some commands and can get replies from Chatbot. After processing the raw text input received by the user, the system will process the sentence and find the desired action by the user. Using the user input, the system will determine whether or not turn on or turn off a device.

5 Proposed System Architecture

The proposed system has a chatbot application to control the devices at home. The mobile application is hosted on Firebase, and it can be accessed from any device with an internet connection. The setup step is used to provide information about the house to the chatbot algorithm. During the setup phase, the chatbot will gather important data such as the number of rooms and appliances in each area in the house. The user can instruct the chatbot to operate all connected appliances during the Use phase. The user's instructions are processed using various Natural Language Processing algorithms in the usage phase. The recommended approaches will be applied to the text when the user provides the chatbot a command. The process of controlling home appliances by chatbot begins with input text in the mobile application. Figure 2 illustrates the proposed system for chatbot.

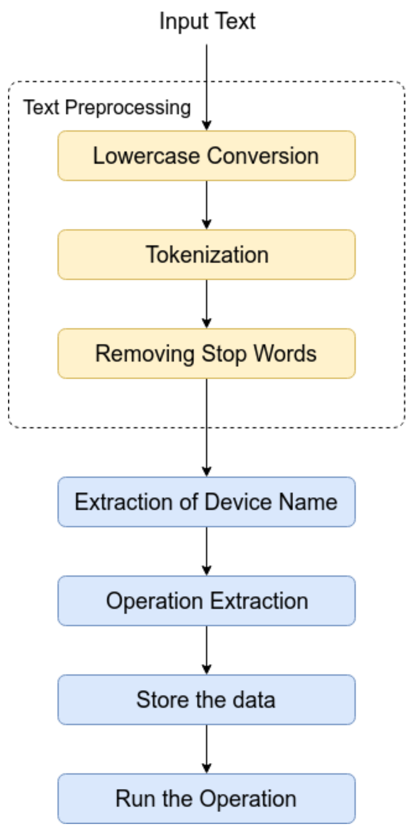

Fig. 2: Chatbot System Architecture

5.0.1 Text Preprocessing

Text can take many different forms such as individual words, multiple paragraphs with special characters. Text preprocessing is a type of data mining that entails converting raw text into a legible format. Real-world data is usually inadequate, inconsistent, or missing in specific behaviors or patterns, as well as being riddled with inaccuracies. In this text preprocessing, We used Natural Language ToolKit (NLTK), which is one of the most well-known and widely-used natural language processing packages in the Python ecosystem. It can be used for a variety of tasks, including stop-word removal, tokenization, and part-of-speech tagging, among others.

5.0.2 Lowercase Conversion

This is the first step of text preprocessing. In this step, the input natural language query is converted into a lower case format. Because there is no difference between uppercase and lowercase forms of words, all uppercase characters are normally transformed to their lowercase counterparts before classification.

5.0.3 Tokenization

Tokens are the building elements of Natural Language, and it is the most popular means of processing raw text. Tokenization is a technique for breaking down paragraphs into phrases or sentences into single words. It used the corpus to generate tokens. To construct a list of different sentences for the former, utilize Sentence Boundary Disambiguation (SBD). This is done using a pre-trained language-specific algorithm like the NLTK's Punkt Models. By using an unsupervised approach to build a model for abbreviated phrases, this tokenizer breaks a text into a series of sentences. It must be trained on a substantial amount of plaintext in the target language before it can be put to use. A pre-trained English Punkt tokenizer is included in the NLTK data package.

5.0.4 Removing Stop Words

Instead of showing themes, objects, or intent, the bulk of words in a text are utilized to link sentence portions. By comparing text to a list of stop words, a term like or and can be deleted. The stop word procedure is illustrated by the algorithm 1.

Algorithm 1: Removing Stop Words

```
Input: W = All the input words; S = List of Stop Words
Output: R: All the words after removing stop words
c = CountWord(W)
for c ∈ C do
  | w = W[c] TOKENS = TOKENIZE(w)
  | for r ∈ TOKENS do
  |   | TOKENS = EMPTY
  |   | if r ∉ S then
  |   |   | PUSH(TOKENS, t)
  |   | end
  | end
  | PUSH(R, TOKENS)
end
return R
```

5.0.5 Extraction of Device Name

Extraction of the device name is a particular solution that uses the Jaro-Winkler string matching algorithm to extract the device name from the input text. Jaro-Winkler Distance formula is used to calculate the distance (d_j) between the two strings A and S.

$$d_j = \frac{1}{3}\left(\frac{n}{\mid A \mid} + \frac{n}{\mid S \mid} + \frac{n-t}{n}\right) \times 100\% \tag{1}$$

where,

- n is the same number of letter
- $|A|$ is the length of Attribute string
- $|S|$ is the length of input string
- t is the amount of Transposition

Jaro-Winkler (d_w) using prefix scale (p) which provides a prefix on a set of strings, with the following formula,

$$d_w = d_j + \left(l_p(1-d_j)\right) \tag{2}$$

where,

- (d_j) is the outcome of S_1 and S_2 string similarity calculations.
- With a maximum of four letters, l is the length of a letter or the same prefix on a string prefix before we discovered the presence of inequality.
- p is a constant scaling factor

The default value for the constant according to Winkler is $p = 0.1$. Using the user input, the system will determine the device name even if any synonyms used. For determining the operation this will use the Jaro-Winkler string matching algorithm to make all devices like fan or bulb. Depending on the user input, it will try to determine the command related to the text. Algorithm 2 illustrates the process.

Algorithm 2: Device Name Extraction

Input: W = List of device names from Database; T = List of device name from WordNet; S = Set of Similar Words
Output: A = Attributes Name; B = device Name

```
t = CountWord(T)
for i ← 1 to t do
    for j ∈ S do
        THRESHOLD = 80
        PERCENTAGE = JARO-WINKLER(S[j],W[j])
        if PERCENTAGE >THRESHOLD then
            PUSH(A[i],W[j])
            PUSH(B[i],T[i])
        end
    end
end
```

5.0.6 Operation Extraction

In the operation Extraction part, we use a Naive Bayes algorithm to extract operation from the input text. It is a set of categories (sub-populations) to which a new observation belongs in machine learning, as determined by a training set of data that comprises observations (or instances) with known category membership. A categorization model attempts to derive conclusions from observed data. Given one or more inputs, a classification model will attempt to predict the value of one or more outputs. Labels that may be added to a dataset are called outcomes. In a supervised model, a training dataset is supplied into the classification algorithm.
The Naive Bayes classifier assumes that the classification features are unrelated. Despite the fact that this assumption is frequently incorrect, examination of the Bayesian classification issue has revealed that there are some theoretical grounds for Naive Bayes classifiers' seemingly unjustifiable effectiveness. Despite the fact that Naive Bayes overestimates the probability of the picked class, because we only use it to make decisions and not to properly forecast real probabilities, the decision is correct, and the model is accurate.
In a text classification challenge, we'll utilize the document's words (or terms/tokens) to assign it to the right group. The following classifier is created using the "maximum a posteriori (MAP)" decision rule:

$$g_{map} =_{g\in G} \left(\log P(g) + \sum_{1\leq k\leq n_d} \log P(t_k|g) \right) \tag{3}$$

As a result, we chose the class with the greatest log score rather than the one with the highest likelihood. Because the logarithm function is monotonic, MAP's judgment remains unchanged.

The final issue to solve is that if a feature/word is absent from a class, its conditional probability is zero. The product of probabilities is 0 if we use the first choice approach; however, the $\log(0)$ is undefined if we employ the second choice technique (sum of their logarithms). To avoid this, we'll smooth each count using add-one or Laplace smoothing:

$$P(t|g) = \frac{L_{T_{gt}} + 1}{\sum_{t' \in V}(L_{T_{gt}} + 1)} = \frac{L_{T_{gt}} + 1}{\sum_{t' \in V}(L_{T_{gt}} + B')} \quad (4)$$

This variation evaluates the conditional probability of a given word/term/token given a class as the relative frequency of term t in texts of group g, according to Multinomial Naive Bayes:

$$P(t|g) = \frac{L_{T_{gt}}}{\sum_{t' \in V} L_{T_{gt'}}} \quad (5)$$

As a result, this variance accounts for the number of instances of word t in group g training papers, including multiple occurrences. In this research, two types of operations are in consideration- on and off. The data set contains words against the appropriate operation. The 80% dataset is considered as the training data set and the input processed text as the test data set. The result will be saved as a .csv format for further use.

5.0.7 Training

The training phase is the first step for classification. The algorithm learns from the data in the training set. In supervised learning problems, each observation includes one or more observed input variables and an observed outcome variable. The binomial classifier is first trained with labeled dataset as this is a supervised approach for classification.

Algorithm 3 represent the Multinomial Naive Bayes Training Algorithm. The training dataset contains sample words. Each word is tagged with the command type of operation. The proposed system is prepared with the training sub-dataset from the dataset. Using the Multinomial Naive Bayes classifier, the sentence predict which class it belongs. The highest conditional probability of output from the sentence is calculated using Bayes theorem. Part of the training data set is represented in Table 1.

5.0.8 Testing

A test data set is also imported as .csv format and the output is also given in the .csv format tagging each comment with the command type it contains. This output also shows the percentage of each type of command content in a specific comment. Algorithm 4 represent the Multinomial Naive Bayes Testing Algorithm.

Algorithm 3: Multinomial Naive Bayes Training Algorithm

Input: I = Input Text; L_T = List of tokens; G = List of group; t_w = training words
$V \leftarrow$ EXTRACTVOCABULARY (I)
$N_T \leftarrow$ COUNTTOKENS (L_T)
for ***each*** $g \in G$ **do**
  $N_g \leftarrow$ COUNTTOKENSINLINES (I, g)
  $priority[g] \leftarrow N_g \setminus N_T$
  $text_g \leftarrow$ JOINTEXTOFALLTOKENSINCLASS (L_T,G)
  for ***each*** $t_w \in V$ **do**
    $L_{T_{gt}} \leftarrow$ COUNTTOKENSOFTERM ($text_g$,t_w)
    for ***each*** $t_w \in V$ **do**
      $cond_{prob}[t_w][g] \leftarrow \frac{L_{T_{gt}}+1}{\sum_{t'}(L_{T_{gt'}}+1)}$
    end
  end
end
return V, *priority, cond*$_{prob}$

Test word	Operation
Start	On
Put On	On
Switch On	On
Turn On	On
Launch	On
Begin	On
Initiate	On
Stop	Off
Put Off	Off
Switch Off	Off
Terminate	Off
Illuminate	Off
Eliminate	Off
Shut down	Off
Cut Off	Off

Table 1: Part of Training Data Set

Table 2 shows the part of the output.

5.0.9 Store the data

If the word is not understandable by any process given on earlier then, the chatbot will ask the user again what he/she wants. We use knowledge base technology to store doubtful words. By this, the system can learn from the user some new words day by day. If the word is not understandable by any process given on earlier then, the chatbot will ask the user again what he/she wants. We use knowledge base tech-

Algorithm 4: Multinomial Naive Bayes Testing Algorithm

Input: G = Group; V = List of Vocabulary; $cond_{prob}$ = conditional probability; d = Documents
E = EXTRACTTOKENSFROMDOCUMENTS(V, d)
for *each* $g \in G$ **do**
  outcome[g] $\leftarrow \log\left(priority[g]\right)$
  for *each* $t \in E$ **do**
    $outcome[g] \mathrel{+}= \log\left(cond_{prob}[t][g]\right)$
  end
end
return $argmax_{g \in C} outcome[g]$

Sentence	On (%)	Off (%)	Result
Terminate the connection of fan	12.40	85.20	Off
Switch on the light	99.36	0.64	On
Switch off the fan	1.32	97.56	Off
Make bright the light	83.71	16.29	On
Shut down the light	2.10	96.70	Off

Table 2: Output for multinomial naive bayes classification

nology to store doubtful words. By this, the system can learn from the user some new words day by day.

5.0.10 Run the Operation

The setup and use phases of the chatbot implementation are basically divided into two sections. The user will be asked for the information needed to set up the chatbot during the setup process. When the chatbot's setting is complete, it will know how many appliances are in each area of the house. During the use phase, users will be able to ask the chatbot to execute a variety of tasks, such as turning on and off lights and fans, as well as setting timers for the same.

6 Result

Running the application for the first time after installation in the mobile phone automatically opens a login activity to connect the user to the online web server. It requires only phone number shown in figure 3. Then the user will get verification code (a six-digit number) known as OTP (One Time Password) in his/her phone. After successfully get the OTP verification code, the user needs to type the code in the designated space. Once the OTP is entered, it is verified and the user will log in into their account as shown in figure 3.

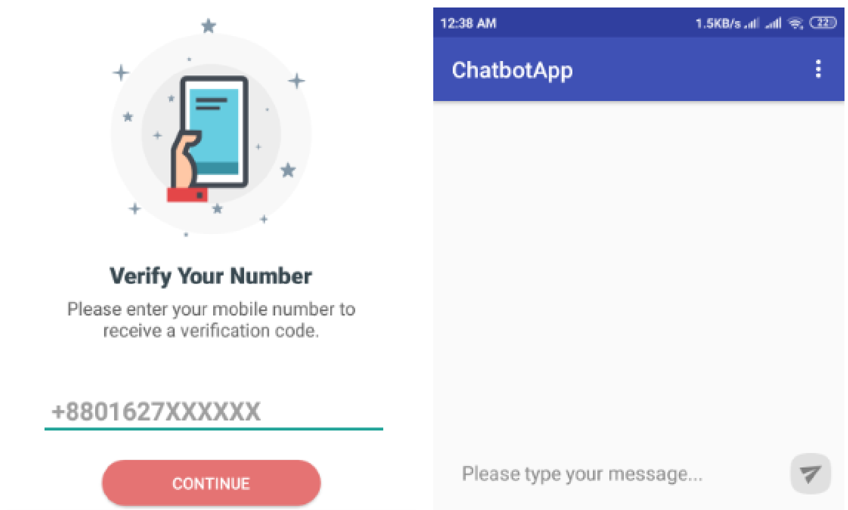

Fig. 3: Login Interface and Chatbot Screen

To control the home appliance user just need to type their command on the app as shown in figure 4. The system will then process the sentence and find the desired action by the user. Using the user input, the system will determine whether or not turn on or turn off a device. If the word is understandable then it will perform what the user wants, which is shown in figure 4. Otherwise, it will ask the user again what actually he/she wants.

The chatbot will always reply to the user. They can control the fan or any other device other than the light bulb. If the user wants to log out they can do it easily. We created a system that control main appliances in a home as part of the project. The user can use a computer or a portable device to engage with the system via a web application or a Chatbot. As a result, it is now easier to handle household appliances with this technology. The Chatbot program may also be used to control household equipment. A web server hosts the Chatbot application. The Chatbot program that has been created is capable of analyzing a text and finding keywords and actions

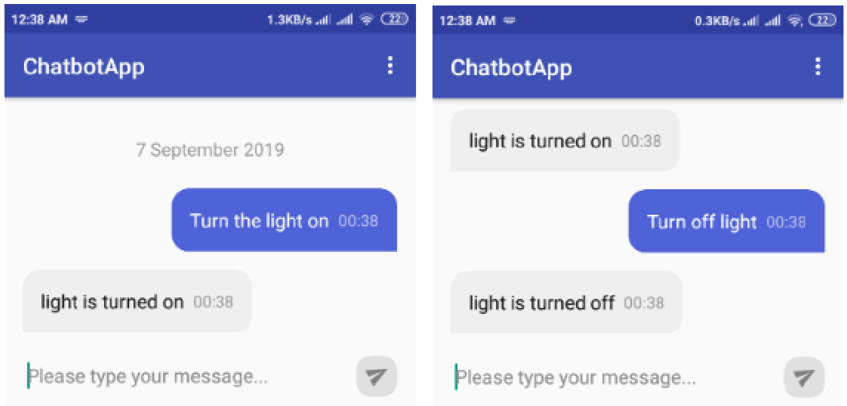

Fig. 4: Light On and Light Off command

from it. It then use this data to carry out the necessary tasks. We can manage the lights, fans, and appliances in the house using the Chatbot program. The accuracy is calculated using the equation 6.

$$Accuracy = \sum_{i=0}^{n} \frac{Prediction_{correct}}{Prediction_{total}} \times 100\% \tag{6}$$

where, $Prediction_{correct}$ is the number of correctly predicted class & $Prediction_{total}$ is the total number of predictions. Error-rate is estimated as equation 7.

$$Error\ Rate = (1 - P) \times 100\% \tag{7}$$

The result shows that the Naive Bayes algorithm can classified 91.45% accurate class detection with an inaccuracy of 8.55%. The results comparing the proposed method with Support Vector Machine is shown in Table 1. It shows that the accuracy of Naive Bayes is better than SVM. If we increase the size of the training data, more accuracy can be obtained. The existence of a large volume of data is always useful for the naive Bayes model.

Table 3: Comparison with different models

Algorithms	Accuracy (%)	Error Rate (%)
Naive Bayes	91.45	8.55
Support Vector Machine	78.76	21.24

The results comparing the proposed method with Naive bayes and Support Vector Machine is shown in Fig. 5. It demonstrates that the accuracy and error rate of Naive Bayes is better or at least competitive than SVM's results.

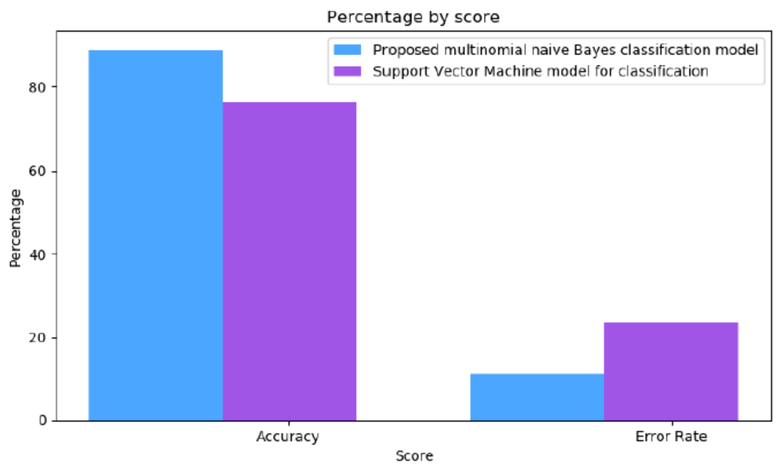

Fig. 5: Comparision Graph

7 Conclusions

Controlling home automation technology remotely is likely to change day to day lifestyle of a user. This article advanced the concept of using chatbots to address some of the challenges of IoT. Because the context is better determined, chatbots will be able to understand the conversation's possibilities. In our proposed system users can control and monitor the home environment by using an android application. The android based home application communicates with the Raspberry Pi 3 via the internet. Any android supported device can be used to install the smart home application. It improves the certain system which supports the lifestyle of the individuals. In the future, different sensor-based control can be added with Chatbot. Self-automation will make this system more user-friendly. Besides home appliances, this system can be used for other gadgets controlling like vehicles. Home security like access control, alarm, etc. can be an extension of this system.

References

1. M.-H. Chao, A. J. Trappey, and C.-T. Wu, "Emerging technologies of natural language-enabled chatbots: A review and trend forecast using intelligent ontology extraction and patent analytics," Complexity, vol. 2021, 2021.
2. A. R. M. Forkan, P. P. Jayaraman, Y.-B. Kang, and A. Morshed, "Echo:A tool for empirical evaluation cloud chatbots," in 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID). IEEE, 2020, pp. 669–672.
3. E. Bezverhny, K. Dadteev, L. Barykin, S. Nemeshaev, and V. Klimov, "Use of chat bots in learning management systems," Procedia Computer Science, vol. 169, pp. 652–655, 2020.
4. B. Leonardo and S. Hansun, "Text documents plagiarism detection using rabin-karp and jaro-winkler distance algorithms," Indonesian Journal of Electrical Engineering and Computer Science, vol. 5, no. 2, pp. 462– 471, 2017.
5. S. Chen, G. I. Webb, L. Liu, and X. Ma, "A novel selective naıve bayes algorithm," Knowledge-Based Systems, vol. 192, p. 105361, 2020.
6. V. Kepuska and G. Bohouta, "Next-generation of virtual personal assistants (microsoft cortana, apple siri, amazon alexa and google home)," in 2018 IEEE 8th annual computing and communication workshop and conference (CCWC). IEEE, 2018, pp. 99–103.
7. M. B. Hoy, "Alexa, siri, cortana, and more: an introduction to voice assistants," Medical reference services quarterly, vol. 37, no. 1, pp. 81–88, 2018.
8. E. Kasthuri and S. Balaji, "Natural language processing and deep learning chatbot using long short term memory algorithm," Materials Today: Proceedings, 2021.
9. [11] J. El Zini, Y. Rizk, M. Awad, and J. Antoun, "Towards a deep learning question-answering specialized chatbot for objective structured clinical examinations," in 2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019, pp. 1–9.
10. F. Patel, R. Thakore, I. Nandwani, and S. K. Bharti, "Combating depression in students using an intelligent chatbot: a cognitive behavioral therapy," in 2019 IEEE 16th India Council International Conference (INDICON). IEEE, 2019, pp. 1–4.
11. A. Sreelakshmi, S. Abhinaya, A. Nair, and S. J. Nirmala, "A question answering and quiz generation chatbot for education," in 2019 Grace Hopper Celebration India (GHCI). IEEE, 2019, pp. 1–6.
12. N. A. Omoregbe, I. O. Ndaman, S. Misra, O. O. Abayomi-Alli, and R. Damaˇseviˇcius, "Text messaging-based medical diagnosis using natural language processing and fuzzy logic," Journal of Healthcare Engineering, vol. 2020, 2020.
13. A. Greene, C. C. Greene, and C. Greene, "Artificial intelligence, chatbots, and the future of medicine." The Lancet. Oncology, vol. 20, no. 4, pp. 481–482, 2019.
14. S. Roca, J. Sancho, J. Garc ıa, and A. Alesanco, "Microservice chatbot architecture for chronic patient support," Journal of biomedical informatics, vol. 102, p. 103305, 2020.
15. R. Pradeep, S. Praveen Kumar, S. Sasikumar, P. Valarmathie, and P. Gopirajan, "Artificial intelligence-based automation system for health care applications: Medbot," in Soft Computing for Security Applications. Springer, 2022, pp. 191–203.
16. P. V. Gopirajan, K. P. Gopinath, G. Sivaranjani, and J. Arun, "Optimization of hydrothermal liquefaction process through machine learning approach: process conditions and oil yield," Biomass Conversion and Biorefinery, pp. 1–10, 2021.
17. V. Vashisht and P. Dharia, "Integrating chatbot application with qlik sense business intelligence (bi) tool using natural language processing (nlp)," in Micro-Electronics and Telecommunication Engineering. Springer, Singapore, 2020, pp. 683–692.
18. N. Rosruen and T. Samanchuen, "Chatbot utilization for medical consultant system," in 2018 3rd technology innovation management and engineering science international conference (TIMES-iCON). IEEE, 2018, pp. 1–5.
19. J. Cho and G. Lee, "A chatbot system for construction daily report information management," in ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction, vol. 36. IAARC Publications, 2019, pp. 429–437.