ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2023.0423
SPECIAL ISSUE ON INNOVATIONS AND INTELLIGENCE FOR COMPUTING AND TECHNOLOGIES (IICT-2023)

BOOSTING THE ACCURACY OF OPTIMISATION CHATBOT BY RANDOM FOREST WITH HALVING GRID SEARCH HYPERPARAMETER TUNING

Bedre Nagaraj[1] and Kiran B. Malagi[2]
[1]*Department of Computer Science and Engineering, Dayanand Sagar University, India*
[2]*Department of Cyber Security, Dayanand Sagar University, India*

Abstract

Computer science, engineering and technologies are witnessing a vital role in providing challenging demands of users. Artificial intelligence, machine learning and robotic process automation strive to improve the intelligent behavior of computers. Fast human like responses of text chatbot can perform better if and only if it is optimized. Hyper parameter optimization methods are popular for successfully boosting up the overall performance of model. In this paper we focus on creating chatbot using random forest and optimizing its performance by hyper parameter tuning halving grid search. We propose chatbot model 1 without optimization, chatbot model 2 with optimization and chatbot model3 with optimization and best values of key performance indicators. Computations are performed before optimization and after optimization for measurement factors including accuracy, precision, recall and f1-scores. Three different models proposed, and performance are compared for each model with respect to precision, recall, f1-scores and accuracy.

Keywords:

Optimization Chatbot, Artificial Intelligence, Machine Learning, Halving Grid Search Hyper Parameter Tuning, Robotic Process Automation

1. INTRODUCTION

Artificial intelligence is significant for developing intelligent machines with learning and problem-solving capabilities. AI is applicable to a wide range of problems [1]. Machine learning efficiently builds machines for complex data handling with an emphasis on adjusting heavy input data, extracting important data, fast knowledge computation, reducing redesign, and making it easy for new knowledge tracking [2] [3]. Chatbots are computer-based programs that imitate human-like interactions with their users (text-to-text). The aim of chatbots is to mimic real human conversations for users [4] - [6].

Different aspects of this work include:

- A study of chatbots reveals that Eliza, Alice, Barry, SmarterChild, Watson, and Mitsuku are popular text chatbots. Most text chatbots have applications in various domains, including the banking sector. Literature has shown a lack of importance to (a) machine learning chatbots (b) comparison and classification of chatbots. (c) evaluation of chatbots [33].
- The work on classifications of chatbots [34] indicates different chatbot classifications: Based on the amount of human assistance, the chatbots are classified into (a) human-mediated chatbots and (b) autonomous/automated chatbots. Based on the communication channel, a chatbot may be classified as (a) text only (b) audio (c) audio and text. Based on AI or non-AI, chatbots are classified as (a) AI chatbots, (b) rule-based chatbots, or (c) mixed-mode chatbots [34].
- We narrow down the focus on text chatbots, AI chatbots, human-mediated chatbots, and autonomous/automated chatbots.
- *Chatbots*: Botpress, Botsify, Botsociety, Botstar, Bot.xo, Chatize, Chatfuel,Chengo, Clustaar, Crisp, Drift, Engati, Flow.xo, Flow.ai, and Freshchat are compared with features such as texting and customer help. All these text chatbots support customer help features [20] [21].

Further, our attention is on customer help, in which chatbots transfer control of user input to human agents. This task may be human mediated (manual) or automated (autonomous). User or customer needs to get ticket in chat. Different goals have different tickets for transferring to different groups of human agents. So in the customer help feature of a text chatbot, the task of generating tickets to map each user query to a human agent will have two scenarios: (i) Automated if performance (percentage of accuracy) is high (high level of accuracy and high level of automation); (ii) Human-mediated (manual) if performance (percentage of accuracy) is low (low level of accuracy and low level of automation).

If the performance of a chatbot is increased, it tends towards higher levels of automation. Therefore, it is essential to boost the accuracy of the chatbot when performing customer service tasks automatically.

GRID search hyperparameter tuning is used in the work [34] for the design and development of a random forest optimization chatbot. The work stated that future research directions can be taken using other hyperparameter tuning methods.

In this paper, we improve the accuracy (which enhances levels of automation) of a designed and developed random forest machine learning bank text chatbot by optimisation with halving grid search hyperparameter tuning.

1.1 MOTIVATION

This study is motivated to get solutions for research questions and to perform different sets of objectives (Section 1.2):

- What are the measurement factors for chatbot performance?
- How can chatbot performance be improved?
- How much w much of the chatbot’s accuracy can be boosted by using optimization?
- What is the role of hyperparameter tuning in the optimization chatbot?
- Why are the best hyperparameters significant?
- How to obtain key performance indicators of an optimized chatbot with high performance
- Why do precision, recall, f1-scores, and accuracy vary from one model to another?

1.2 OBJECTIVES

Our study has the following objectives:

- Design and development of an optimized chatbot with random forest and halving grid search hyperparameter tuning.
- Propose model for optimization chatbot with halving grid search and hyperparameter tuning.
- Evaluate the performance of the developed chatbot.
- Improve the performance of developed chatbot models 1 and 2 by optimization and hyperparameter tuning to get an optimized chatbot.
- Explore and compare different evaluation factors for chatbots.
- Identify the role of hyperparameter tuning and key performance indicators.
- Select the key performance indicators on which optimized chatbot performance depends. Measure the precision, recall, f1-score, and accuracy of models 1 and 2 (i.e., before and after optimization).

1.3 CONTRIBUTION

The contributions to this work are as follows:

- The creation of a random forest chatbot, evaluation, and optimization using hyperparameter tuning halved grid search optimization. Identification of key performance indicators and optimizing them using halved grid search
- This work contributes three different chatbot models: model 1 without optimization, model 2 with optimization, and model 3 with identification of key performance indicators and optimization.
- This work has made a major contribution to boosting the accuracy (by 3.57%) of an optimized chatbot.
- Also, the precision, recall, and f1 scores of Models 3 are enhanced as compared to Model 1 or Model 2.
- This is the first work that has performed key performance indicator identification and applied hyperparameter tuning optimization to it.

This paper is organized as follows: section 2: literature review; section 3: proposed methodology; section 4: results and discussion, and section 5: conclusion and future enhancements.

2. LITERATURE REVIEW

Robotic process automation automates manual tasks using robots [7] [8]. Different chatbot tasks are distributed to resources based on dependencies. Chatbot automation capabilities are powered by artificial intelligence and learning algorithms with support to overcome restrictions on dependencies of automation capabilities [9]-[12].

Machine learning algorithms can be successfully implemented by important feature hyperparameter tuning [13]. The importance of hyperparameter tuning is presented [14] as useful in enhancing the model's performance.

The work [15] shows how to perform hyperparameter tuning to select the best hyperparameter values by walking through every hyperparameter space and computing the accuracy. Decision trees used in this work [16] to compute accuracy before and after optimization. Optimized random forest [17], used in another study with tenfold cross validation, has shown an increase in performance levels. The work [18] with ensemble methods used for prediction by obtaining hyperparameter best values using PCA and LDA also proved to affect accuracy levels. A chatbot model is proposed with resource types, resource characteristics, and levels of automation; for chatbot intent classification, logistic regression is used, and for process automation, random forest is used [19]. Another work proposed an optimization chatbot model for the evaluation of 16 different chatbots along with an algorithm to predict levels of chatbot optimization. [20][21]. Grid search hyperparameter tuning is a brute-force algorithm used for easy and parallel execution of exhaustive search walking through hyperparameter space [22]. The other works used random search [23], Bayesian optimization [24], gradient optimization [25], support vector machines (SVM) [25], gradient boost (GB) [26], extra tree (ET) [27], K-nearest neighbor (KNN) [28], decision tree (DT) [29], AdaBoost (AB) [30], random forest [31], and logistic regression.

2.1 RESEARCH GAPS

We have identified various research gaps from the literature: Chatbot with a random forest classifier has an accuracy of 72.6% and needs optimization. Explore and compare different classifiers in developing chatbots and select the best. KPI-based evaluation of the chatbot is necessary. Use other hyperparameter tuning methods. [19] [20] [21] [32]

3. PROPOSED METHODOLOGY

In this section, we consider the role of hyperparameter tuning and key performance indicators. We proposed an optimized chatbot model and designed an algorithm for developing an optimized chatbot using random forest and hyperparameter tuning optimization.

3.1 PROPOSED OPTIMISATION CHATBOT MODEL

An optimized chatbot model is proposed to consist of a set of activities, a set of constraints, a set of resource characteristics, levels of automation, resource types, a set of parameters (features), a set of hyperparameters, and key performance indicators, a random forest classifier, a hyperparameter tuning method (halving grid search), a set of inputs, and a set of outputs.

Each activity set has one or more activities. Each activity has a label, a set of one or more resources permitted to perform the activity, input, and output. *N* and *S* represent int and string types, respectively in the input and output. Each resource has a role, a resource type (human agent, chatbot), and a set of resource characteristics. If the performance of the chatbot is low on a task or incident, then a human agent will execute the task without automation (manual); if the chatbot has high performance, then chatbot tasks are automated. Tasks in a process are either manual (performed by human participants) or supported automatically by a chatbot (executed by software) [32].

Hyperparameters are set to get a more relevant optimized model. The importance of hyperparameter tuning is to minimize errors and increase the convergence of machine learning algorithms to a greater extent.

Now we develop an optimized chatbot for detecting the best team for providing the required service using the business process event log of the banking dataset [19], which has interaction management and incident management. This chatbot identifies the team to handle tasks and assigns them to them automatically. Inputs to the classifier are the CI name, the CI type, and the component of the incident. Team of incidents is output. Precession, recall, and accuracy are performance measures that are enhanced using hyperparameter tuning and optimization.

3.2 RANDOM FOREST

The random forest algorithm works on the principle of a bootstrap sample (train data) and an out-of-bag sample (test data) (steps 1 and 2). Bagging is performed by introducing randomness to avoid the correlation among different decision trees (step 3). As in step 4, we use averaging for regression tasks and voting for classification tasks. Step 5 is used to perform the cross validation across the out-of-bag sample, i.e., unseen data [432].

3.3 PROPOSED ALGORITHM

Now we focus on creating, evaluating, and enhancing the chatbot, resulting in an optimized chatbot with high performance and the corresponding algorithm proposed as follows:

Algorithm: design and development of a random forest chatbot and optimization of the chatbot using hyperparameter tuning and halving grid search (chatbot model 1 refers to a random forest chatbot without optimization). Chatbot model 2 refers to a random forest chatbot with hyperparameter tuning optimization using a halving grid search method.

Step 1: Read the input file (data set) and perform pre-processing. to prepare the data for a machine learning classifier.

Step 2: Refined data set split into seen and unseen data (test data)

Step 3: Compute the performances of chatbot model1 and store values of the macro average precision, macro average recall, macro-average f1-scores, weighted average precision, weighted average recall, weighted average f1-scores and accuracy in the variables p_1, r_1, f_1, p_2, r_2, f_2, and acc, respectively.

Step 4: Compute the performances of chatbot model 2 and store the values of macro-average precision, macro-average recall, macro average f1-scores, weighted average precision, weighted average recall. weighted average f1-scores and accuracy in the variables p_1hpt and r_1hpt, f_1hpt, p_2hpt, r_2hpt, f_2hpt, and $acchpt$, respectively.

Step 5: Compute the performances of chatbot model 3 and store the macro-average precision, the macro-average recall, macro average f1-scores, weighted average precision, weighted average recall. weighted average f1-scores and accuracy in the variables p_1kh, r_1kh, and f_1kh, p_2kh, r_2kh, f_2kh, and $acckh$, respectively.

Step 6: Identify the percentage of boosting the performance. Evaluation factors by Model 2 with computations:

$$m_2p_1=p_1hpt-p_1;$$
$$m_2r_1=r_1hpt-r_1;$$
$$m_2f_1 = f_1hpt-f_1; \tag{1}$$
$$m_2p_2=p_2hpt-p_2,$$
$$m_2r_2=r_2hpt-r_2,$$
$$m_2f_2=f_2hpt-f_2, \tag{2}$$
$$m_2acc= acchpt-acc. \tag{3}$$

where m_2p_1, m_2r_1, m_2f_1, m_2p_2, m_2r_2, m_2f_2, and m_2acc denote enhanced values of macro average precision, macro average recall, macro average f1 score, weighted average precision, weighted average recall, weighted average f1 score, and accuracy by model 2,

Step 7: Compute the percentage of boosting the performance. Evaluation factors for model 3 over model 2:

$$m_3p_1=p_1kh-p_1hpt;$$
$$m_3r_1=r_1kh-r_1hpt;$$
$$m_3f_1 = f_1kh-f_1hpt; \tag{4}$$
$$m_3p_2=p_2kh-p_2hpt,$$
$$m_3r_2=r_2kh-r_2hpt,$$
$$m_3f_2=f_2kh-f_2hpt, \tag{5}$$
$$m_3acc= acckh-acc. \tag{6}$$

where m_3p_1, m_3r_1, m_3f_1, m_3p_2, m_3r_2, m_3f_2, m_3acc denote enhanced values of macro average precision, macro average recall, macro average f1 score, weighted average precision, weighted average recall, weighted average f1 score, and accuracy, respectively, by model 3 over model 2,

Step 8: Compute the percentage of boosting the performance. Evaluation factors for model 3 over model 1:

$$mp_1=p_1kh-p_1;$$
$$mr_1=r_1kh-r_1;$$
$$mf_1 = f_1kh-f_1 \tag{7}$$
$$mp_2=p_2kh-p_2;$$
$$mr_2=r_2kh-r_2;$$
$$mf_2=f_2kh-f_2; \tag{8}$$
$$macc= acckh-acc. \tag{9}$$

where mp_1, mr_1, mf_1, mp_2, mr_2, mf_2, $macc$ denote enhanced values of macro average precision, macro average recall, macro average f1 score, weighted average precision, weighted average recall, weighted average f1 score, and accuracy, respectively, by model 3 over model 1.

3.4 DISCUSSION OF THE PROPOSED ALGORITHM

Now we will consider the discussion of the proposed algorithm. Step 1 includes reading the data set (.csv file read wrt banking BPIC2014 log), extracting the relevant information (and transforming by one hot encoding the relevant information (and transforming by one hot encoding), i.e., features, and preparing the data for a machine learning classifier, and preparing the data for a machine learning classifier. This includes the importing of the packages supported by Python like numpy, spacy, panda, etc. In step 2, the entire refined data is split into two parts: training data and test data. Train data is used to build the chatbot, and test data is used as unseen data for the learning and testing phases of the chatbot. Step 3 is for the Model 1 of the chatbot without any optimization, which is considered for finding performance factors like the macro average and weighted average parameters of precision, recall, and F1 score. Also, the accuracy is computed for Model 1. Similarly, in step 4 for model 2, i.e., the chatbot with hyperparameter tuning and halved grid search optimization, the values are computed and stored. For model 3, which is the enhanced version having key performance indicators and halved grid search, the values are computed and stored in variables as mentioned in step 5. In step 6, all parameters are computed, which shows enhancements by model 2 over model 1. The next step computes enhanced values of performance parameters for model 3 over model 2. Now, finally, we find the enhancement factors of our Model 3 over our Model 1.

Model 1 has only a chatbot model with a random forest without any optimization. Model 2 represents a random forest chatbot model with hyperparameter tuning and halving grid search to select the best parameters to get the highest accuracy. Whereas in the Model 3, we also use the key performance indicators along with hyperparameter tuning with a random forest optimized chatbot and halving grid search to boost accuracy.

3.5 MATHEMATICAL MODELING

The optimization chatbot mathematical model has an input (data from a test set during performance assessment) with a 1:1 mapping to a distinct output (accuracy). The various parameters such as min_samples_leaf, min_samples_split, max_depth, max_features, and n_estimators are considered functions that transform each possible input from test data into distinct output, i.e., accuracy.

Let f be a function with hyperparameters to bind input a to output b, where a is input from the test dataset and b is accuracy. $f: a \rightarrow b$, where a and b are distinct inputs and outputs, respectively. This mathematical modeling is used to simulate the proposed algorithm on various models, as shown in blocks of Fig.1-Fig.3, which correspond to Models 1, 2, and 3.

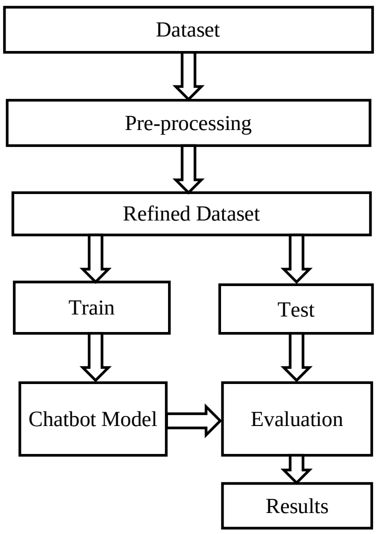

Fig.1. Proposed Chatbot model 1 without any Optimisation

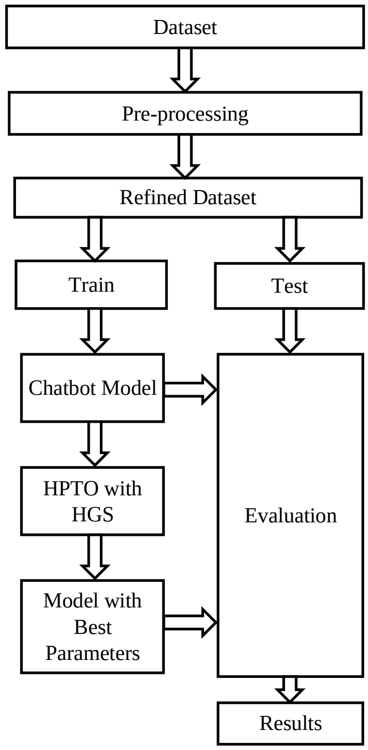

Fig.2. Proposed Optimization Chatbot model2 with Hyper Parameter tuning Halving Grid Search Optimization

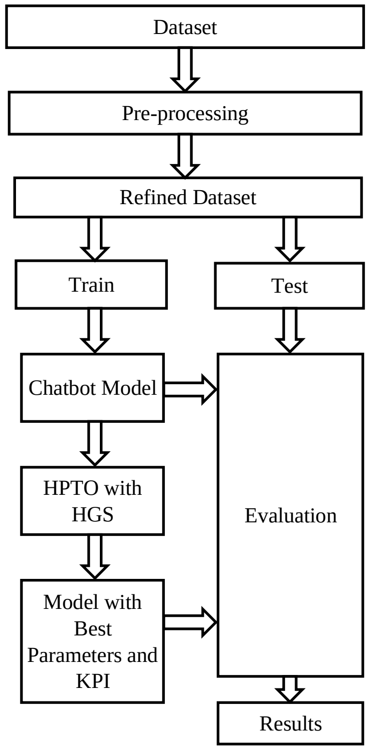

Fig.3. Proposed Optimization Chatbot Model3 with Hyper Parameter Halving Grid Search Optimization and Key Performance Indicators (KPI)

3.6 SIMULATION TOOLS FOR THE PROPOSED ALGORITHM

The algorithm presented in Section 3.3 is implemented using simulation tools in Python with Scikit-Tools.

Python is a user-friendly, robust, platform-independent, portable language with a rich set of support for the simulation of machine learning algorithms and for hyperparameter tuning. The sklearn.ensemble.RandomForestClassifier() supports the usage of the random forest classifier, which is one of the ensemble methods provided by Sklearn to simulate the random forest classifier. Param_grid is used to define the set of parameters needed in order to perform a halving grid search. This parameter grid contains values for bootstrap, max_depth, max_features, n_estimators, min_samples_split, min_samples_leaf, etc. The sample parameter grid form is as follows:

param_grid =

“bootstrap”: [True], “min_samples_leaf”: [3, 4, 5],

'min_samples_split': [8, 10, 12].

'max_depth': [30, 240, 275],

'max_features': [110, 200, 250], 'n_estimators': [400, 500]

}

HalvingGridsearchCV is used to perform a halving grid search. HalvingRandomisedSearchCV (estimator = rf, param_grid = param_grid) is used to find the best parameters among the grid parameters specified. Among these parameters, some, such as max_depth and max_features, can be considered key performance indicators, and the halving grid search is continued to simulate the model. Implemented results are discussed in section 4.

3.7 SCOPE OF THE PROPOSED METHODOLOGY

The scope of this work is to “design an algorithm and develop a chatbot using random forest, evaluate performance, and enhance performance by hyperparameter tuning optimization-halving grid search with identification of key performance indicators responsible for optimisation chatbot’s highest performance”.

4. RESULTS AND DISCUSSION

The implementation of the proposed algorithm (section 3) using the Python library SciKit tool resulted in the accuracy of a random forest chatbot of 73.38% (accuracy acc1) and the accuracy of an optimization chatbot of 74.96% with hyperparameter tuning. We observed that the optimization process by hyperparameter tuning enhanced the accuracy by 1.58%. This scenario has two models of chatbot optimization: random forest and halving grid search during hyperparameter tuning.

First, a chatbot model with random forest without optimization Second, a chatbot model with random forest using hyperparameter optimization halving grid search in model 2, halving grid search, hyperparameter tuning is applied to the random forest chatbot for optimization with parameters max_depth, max_features, min_samples_leaf, min_samples_split, and n_estimators. The precision, recall, f1-score, and accuracy values computed for different values for the above parameters and values are as in table 1 and table 2 (plots as in Fig.4 and Fig.5), respectively, for model 1 and model 2.

Further, the experiment is conducted with Model 2 to detect the KPI (key performance indicator) values. These kpi values are found to be maxdepth (110), maxfeatures (100), nestimators (1), minsample leaf (4), minsample split (2), and randomstate (1). The Fig.6 depicts a plot of the various performance factors of models 1, 2, and 3. Similar type of work carried out by [19] obtained the accuracy of chatbots at 72.6%. However, for the same dataset, our work yielded an optimized chatbot accuracy of 74.95%. The optimized chatbot of this paper has increased accuracy by 2.35% with the best selection of optimal hyperparameters, i.e., hyperparameter tuning halving grid search, applied appropriately to enhance accuracy.

The Table.3 shows different values obtained for kpi inputs to the model such as maxdepth (90), maxfeatures (291), random state (77), and nestimators (100), This model3, i.e., chatbot optimization model with halving grid search and selected KPI, has the accuracy of 76.9550748%. The accuracy of work [19] with grid search is 72.6%, which is enhanced by our model 3 by a factor of 4.3550748% (76.96 - 72.60). The Fig.4 depicts a graph of the accuracy, precision, recall, and f1scores of models 1, 2, and 3. The Table.4 - Table.6 show the enhancements of model performances. The enhanced precision, recall, F1-score, and accuracy values of chatbot model 2 over model 1 are as in Table.4. Enhanced precision, recall, F1-score, and accuracy values of chatbot model 3 over model 2 are listed in Table.5. Enhanced precision, recall, F1-score, and accuracy values of chatbot model 3 over model 1 are indicated in Table.6. As in Table.4, model 2 has better accuracy by a factor of 1.5807 than model 1. Model 2 has better macro average precision, better weighted average precision, and a better weighted average f1 score by 0.01.

Table.1. Precision, recall, F1-score and accuracy values of chatbot model1 without optimization

Macro Avg			**Weighted Avg**			**Accuracy (acc)**
p1	**r1**	**f2**	**p2**	**r2**	**f2**	
0.65	0.7	0.67	0.74	0.73	0.72	73.38

Table.2. Precision, recall, F1-score and accuracy values of chatbot model2 with optimization

Macro Avg			**Weighted Avg**			**Accuracy (acchpt)**
p1hpt	**r1hpt**	**f1hpt**	**p2hpt**	**r2hpt**	**f2hpt**	
0.66	0.7	0.67	0.73	0.75	0.73	74.96

Table.3. Precision, recall, F1-score and accuracy values of chatbot model3 with optimization and selected kpi from hyperparameters

Macro Avg			**Weighted Avg**			**Accuracy (Acckh)**
p1kh	**r1kh**	**f1kh**	**p2kh**	**r2kh**	**f2kh**	
0.68	0.73	0.69	0.77	0.77	0.76	76.96

Table.4. Enhanced precision, recall, F1-score and accuracy values of chatbot model2 over model 1

Macro Avg			Weighted Avg			Accuracy (m2acc)
m2p1	m2r1	m2f2	m2p2	m2r2	m2f2	
0.01	0	0	0.01	-0.02	0.01	1.5807

Table.5. Enhanced precision, recall, F1-score and accuracy values of chatbot model3 over model2

Macro Avg			Weighted Avg			Accuracy (m3acc)
m3p1	m3r1	m3f2	m3p2	m3r2	m3f2	
0.02	0.03	0.02	0.04	0.02	0.03	1.9966

Table.6. Enhanced precision, recall, F1-score and accuracy values of chatbot model3 over model1

Macro Avg			Weighted Avg			Accuracy (macc)
mp1	mr1	mf2	mp2	mr2	mf2	
0.03	0.03	0.02	0.03	0.04	0.04	3.5773

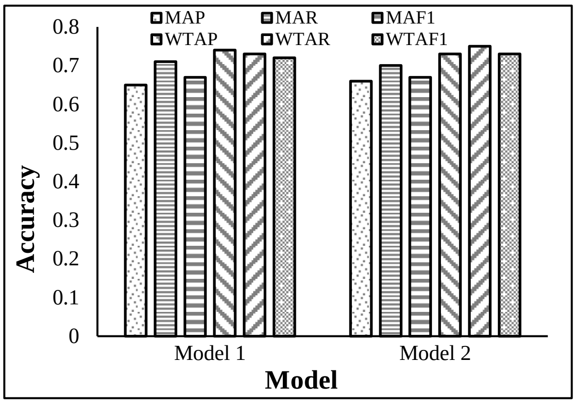

Fig.4. Precision, recall and f1scores of model 1 and model 2

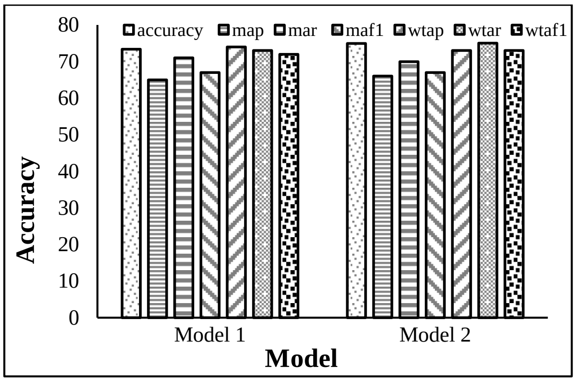

Fig.5. Accuracy, Precision, recall and f1scores of model 1 and model 2

In comparison to Model 2, Model 3 has good accuracy by 1.9966. Model 3 also shows better macro-average precision, macro-average f1 score, and weighted average recall by 0.02; Model 3 has enhanced 0.03% of macro-average recall and weighted average f1 scores.

But when we consider model 1 and model 3, we can observe 3.5773% better accuracy in our model 3. This work shows that the accuracy of chatbots can be enhanced by hyperparameter tuning. As we see from Table.6, accuracy is boosted by the best factor if we consider Model 3, i.e., if we consider key performance indicators and hyper parameter tuning also.

The proposed algorithm has resulted in the optimization of the chatbot through hyperparameter tuning and the identification of key performance indicators, which have not been done by any of the existing literature research works.

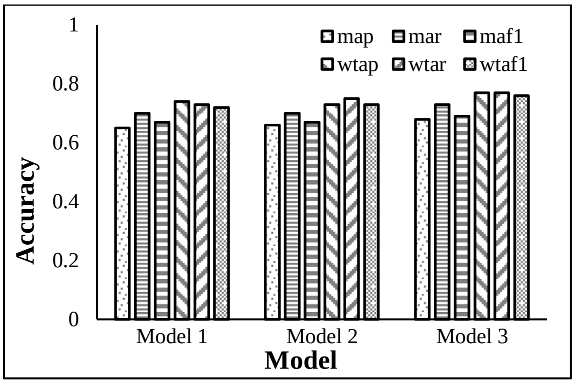

Fig.6. Accuracy of model 1, model 2 and model 3

5. CONCLUSION AND FUTURE ENHANCEMENTS

This work was successful in creating, evaluating, and improving the accuracy of a random forest chatbot, as well as providing all solutions to research questions. An efficient optimized chatbot model has been proposed, as has an algorithm to design and develop an optimized chatbot using the random forest method with hyperparameter tuning. Results are presented and accuracy is boosted. Further, the significant factors, i.e., key performance indicators, are identified, which are responsible for obtaining higher accuracy from the optimization chatbot.

The highest accuracy achieved by this optimization chatbot is 74.95%. In comparison to the work [19] on hyperparameter tuning, we have improved the accuracy of the optimized chatbot by 2.35%. Our Model 3 has better performance by a factor of 4.3550748%. This work can be extended with other hyperparameter optimization methods. Another area for future work is to use a different classifier instead of the random forest method.

REFERENCES

[1] D. Acemoglu and P. Restrepo, "The Race between Machine and Man: Implications of Technology for Growth, Factor Shares and Employment", NBER Working Paper No. 22252, 2016.

[2] M. Boden and S. Kember, "Principles of Robotics: Regulating Robots in the Real World", *Connection Science*, Vol. 29, pp. 124-129, 2017.

[3] V.C. Muller and N. Bostrom, "Future Progress in Artificial Intelligence: A Survey of Expert Opinion", *Proceedings of International Conference on Fundamental Issues of Artificial Intelligence*, pp. 553-571, 2016.

[4] B. Kohli and P. Kumar, "A Platform for Human-Chatbot Interaction using Python", *Proceedings of International*

Conference on Green Computing and Internet of Things, pp. 439-444, 2018.

[5] Chen Wei, Zhichen Yu and Simon Fong, "How to Build a Chatbot: Chatbot Framework and its Capabilities", *Proceedings of International Conference on Machine Learning*, pp.369-373, 2018.

[6] E. Adamopoulou and L. Moussiades, "An Overview of Chatbot Technology", Proceedings of International Conference on Advances in Information and Communication Technology, pp. 1-7, 2020.

[7] Mary Lacity, Leslie Willcocks and Andrew Craig, "Robotic Process Automation at Telefónica O2", *The Outsourcing Unit Working Research Paper Series*, Vol. 15, No, 2, pp. 1-178, 2015.

[8] Leslie Willcocks, Mary Lacity and Andrew Craig, "The IT Function and Robotic Process Automation", *The Outsourcing Unit Working Research Paper Series*, Vol. 15, No. 5, pp. 1-112, 2015.

[9] M. Dumas and H.A. Reijers, "*Fundamentals of Business Process Management*", Springer, 2013.

[10] R. Parasuraman and C.D. Wickens, "A Model for Types and Levels of Human Interaction with Automation", *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, Vol. 30, No. 3, pp. 286-297, 2000.

[11] M. Vagia and S.A. Fjerdingen, "A Literature Review on the Levels of Automation during the Years What are the Different Taxonomies that have been Proposed?", *Applied Ergonomics*, Vol. 53, pp. 190-202, 2016.

[12] A. Jimenez-Ramirez, I. Barba and C. Del Valle, "A Method to Improve the Early Stages of the Robotic Process Automation Lifecycle", *Lecture Notes in Computer Science*, Vol. 11483, pp. 446-461, 2019.

[13] C. Thornton and H. Hoos, "Auto-Weka: Combined Selection and Hyperparameter Optimization of Classification Algorithms", *Proceedings of International Conference on Knowledge Discovery Data Mining*, pp. 847-855, 2013.

[14] P. Probst, B. Bischl and A.L. Boulesteix, "Tunability: Importance of Hyperparameters of Machine Learning Algorithms", *Proceedings of International Conference on Machine Learning*, pp. 1-5, 2018.

[15] B. Shekar and G. Dagnew, "Grid Search-Based Hyperparameter Tuning and Classification of Microarray Cancer Data", *Proceedings of International Conference on Computer Communications*, pp. 1-8, 2020.

[16] R.G. Mantovani, "Hyper-Parameter Tuning of a Decision Tree Induction Algorithm", *Proceedings of International Conference on Intelligent Systems*, pp. 37-42, 2016.

[17] C. Fawcett and H.H. Hoos, "Analyzing Differences between Algorithm Configurations through Ablation", *Journal of Heuristics*, Vol. 22, No. 4, pp. 431-458, 2016.

[18] C. Anisha and N. Arulanand, "Early Prediction of Parkinson's Disease (PD) using Ensemble Classifiers", *Proceedings of International Conference on Innovation Trends in Information Technology*, pp. 1-6, 2020.

[19] R. Sindhgatta, A.H.M. Hofstede and A. Ghose, "Resource Based Adaptive Robotic Process Automation", *Lecture Notes in Computer Science*, pp. 451-466, 2020.

[20] B. Nagaraj and K.B. Malagi, "Research Paper to Design and Develop an Algorithm for Optimization Chatbot", *Proceedings of International Conference on Ubiquitous Intelligent Systems Smart Innovation, Systems and Technologies*, pp. 1-8, 2022.

[21] Bedre Nagaraj and Kiran B. Malagi, "Research Paper to Design and Develop an Algorithm for Optimization Chatbot", *Proceedings of International Conference on Ubiquitous Intelligent Systems*, pp. 1-12, 2022.

[22] M.A. Zoller and M.F. Huber, "Benchmark and Survey of Automated Machine Learning Frameworks", *Proceedings of International Conference on Machine Learning*, pp. 1-6, 2019.

[23] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization", *Journal of Machine Learning Research*, Vol. 13, No. 1, pp. 281-305, 2012.

[24] J. Snoek, H. Larochelle and R.P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms", *Proceedings of International Conference on Machine Learning*, pp. 1-7, 2012.

[25] O. Chapelle, V. Vapnik and O. Bousquet, "Choosing Multiple Parameters for Support Vector Machines", *Machine Learning*, Vol. 46, pp. 1-3, 2002.

[26] J.H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine", *Annals of Statistics*, Vol. 29, No. 5, pp. 1189-1232, 2001.

[27] P. Geurts, D. Ernst and L. Wehenkel, "Extremely Randomized Trees", *Machine Learning*, Vol. 63, No. 1, pp. 3-42, 2006.

[28] S. Zhang and M. Zong "Efficient KNN Classification with Different Numbers of Nearest Neighbors", *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 29, No. 5, pp. 1774-1785, 2017.

[29] M. Reif and A. Dengel, "Prediction of Classifier Training Time including Parameter Optimization", *Proceedings of International Conference on Conference on Artificial Intelligence*, pp. 260-271, 2011.

[30] R. Gao and Z. Liu, "An Improved Adaboost Algorithm for Hyperparameter Optimization", *Journal of Physics Conference Series*, pp. 1-13, 2020.

[31] P. Probst, M.N. Wright and A.L. Boulesteix, "Hyperparameters and Tuning Strategies for Random Forest", *Wiley Interdisciplinary Review*, Vol. 9, No. 3, pp. 1301-1314, 2019.

[32] Bedre Nagaraj and Kiran B. Malagi., "Design and Development of an Optimization Chatbot based on Random Forest Machine Learning Algorithm with Hyper parameter Tuning", *Indian Journal of Natural Sciences*, Vol.13, No. 76, pp. 1-14, 2023.

[33] Bedre Nagaraj and Kiran B. Malagi, "Review of Optimization Chatbot Trends, Tools, Technologies and Research Issues", *Proceedings of International Conference on Innovation Trends in Information Technology*, pp. 1146-1160, 2021.

[34] Bedre Nagaraj and Kiran B. Malagi, "*Design of an Optimization Chatbot Model for Classification Framework*", IFERP Publisher, 2021.