**TITLE**

**Role of Artificial Intelligence in Solving Missing Persons Using K-Nearest Neighbor Algorithm and Comparing With Long short term memory Algorithm**

R.Rupasridevi1, C. Clement Raj2

R.Rupasridevi¹

Research Scholar,

Department of Computer Science and Engineering,

Saveetha School of Engineering,

Saveetha Institute of Medical And Technical Sciences,

Saveetha University, Chennai, Tamil Nadu, India, Pincode: 602105.

[ravurisridevi0092.sse@saveetha.com](mailto:ravurisridevi0092.sse@saveetha.com)

Clement Raj C

Project Guide, Corresponding Author,

Department of Computer Science and Engineering,

Saveetha School of Engineering,

Saveetha Institute of Medical And Technical Sciences,

Saveetha University, Chennai, Tamil Nadu, India, Pincode: 602105.

[clementraj.sse@saveetha.com](mailto:clementraj.sse@saveetha.com)

**Keywords:** Real-Time Tracking, Geographic Information Systems (GIS), Cognitive Computing, Emergency Response, Decision Support System, Neural Networks

**ABSTRACT**

**Aim:** The important goal of this research experiment is to focus on the importance of role of artificial intelligence in solving missing persons using the K-Nearest Neighbor (K-NN) algorithm and compare it with the Long short term memory (lstm) **Materials and Methods:** The K-Nearest Neighbor is used with 20 sets as sample size, and Long short term memory (lstm) has been used with a sample mean size of 20 sets with a total of 40 sets being compared to improve the accuracy of the present research. The mean accuracy of the present research has been calculated using the ClinCalc software appliance under supervised learning with 0.8 as the alpha value, a G-Power value of 0.8, and CI of 95%**.Results:** After performing this research, theK-Nearest Neighbor has attained an accuracy of 87.20% and the Long short term memory (lstm) has achieved an accuracy of 75.25%. An Independent samples T-Test analysis has been executed, and its significance value is found to be p=0.000 (p<0.05), suggesting statistical significance**. Conclusion:** In this present research, K-Nearest Neighbor the algorithm is collated with the long short term memory (lstm). After performing the current research experiment, the K-Nearest Neighbor has been found to have more perfection than the Long short term memory (lstm)

**Keywords:** Real-Time Tracking, Geographic Information Systems (GIS), Cognitive Computing, Emergency Response, Decision Support System, Neural Networks

Ethical Considerations in AI

**INTRODUCTION**

The K-Nearest Neighbor algorithm is a well-established machine learning technique known for its simplicity and effectiveness in classification tasks. Applied to the context of missing persons, KNN leverages spatial and temporal features to identify potential locations where a person may be found based on the characteristics of known data points. This approach can significantly reduce search areas and improve the chances of a successful recovery. On the other hand, the Long Short-Term Memory algorithm, a type of recurrent neural network (RNN), excels in handling sequential data and is particularly adept at capturing dependencies over extended periods. By considering the temporal aspects of missing person’s cases, LSTM can enhance prediction accuracy by incorporating the sequence of events leading up to an individual's disappearance. Endeavors to explore and compare the effectiveness of the K-Nearest Neighbor algorithm and the Long Short-Term Memory algorithm in the context of solving missing person’s cases. By delving into their respective strengths and limitations, we aim to shed light on the most suitable algorithmic approaches for different scenarios. Such insights have the potential to revolutionize search and rescue operations, providing law enforcement agencies and humanitarian organizations with advanced tools to expedite the resolution of missing person’s cases, ultimately contributing to the enhancement of public safety and well-being.

The research papers are collected from the last 5 years i.e.; 2018-2022 and almost 300 research articles have been reported in “IEEE Xplore” and all over 390 research papers are published in the “Science Direct” on Finding Missing Persons Using AI. The IEEE Explore and Science Direct are considered as the main databases in collecting the research papers for this research experiment. Ultimately, this research seeks to offer insights into the applicability and effectiveness of AI-driven approaches in aiding law enforcement agencies, search and rescue teams, and relevant authorities in their efforts to locate missing persons. The findings aim to contribute to the discourse on leveraging technology for societal welfare, potentially facilitating the reunion of missing individuals with their families and promoting community safety and well-being-NN, a foundational machine learning algorithm, operates based on the principle of similarity assessment. In the context of missing persons, K-NN analyzes facial features or biometric data to identify similarities with known individuals from a database, aiding in potential matches or identifications. Conversely, Long short term memory, a subset of deep learning models, excel in learning intricate patterns and representations directly from image data. With their prowess in image analysis and feature extraction, LSTMs hold promise in recognizing missing individuals based on visual cues.

The accuracy of the present research is not exactly existing in the system. The present research experiment is done to improve the accuracy of the K-Nearest Neighbor algorithm and contrast it with the existing algorithm. K-Nearest Neighbor algorithm has been introduced to improve the accuracy of finding missing persons using ai and it is compared with the long short term memory. The advantages and disadvantages of both Machine Learning algorithms have been considered for the current research. In the present article the K-Nearest Neighbor algorithm is compared with the long short term memory algorithm for finding missing persons using ai.

**MATERIALS AND METHODS**

The current experimentation work has been carried out in the Machine Learning Laboratory at Saveetha School of Engineering, Saveetha Institute of Medical And Technical Sciences (SIMATS), Chennai. The sample size has been calculated using the ClinCalc tool under supervised learning with an alpha value of 80% or 0.8 and G-Power value of 0.8 and with a significance value of 0.05 at Confidential Interval (CI) of 95%. The sample size of 20 sets has been used for both the Group 1 i.e; K-Nearest Neighbor algorithm and Group 2 i.e.; long short term memory algorithm, with a total of 40 sets being considered for this research (GE Xinliang, YANG Jie)

**K-Nearest Neighbor algorithm**

K-NN is straightforward to understand and implement, making it an ideal starting point for beginners in machine learning. Its intuitive nature stems from the principle of similarity measurement instances are classified based on their proximity to other instances in the dataset. It can be applied to both classification and regression tasks. In classification, it assigns a class label to an unseen instance based on the majority vote of its K nearest neighbors. In regression, it predicts a continuous value based on the average or weighted average of the K nearest neighbors.it doesn't involve a separate training phase; it stores all the training data instances, making predictions directly from the stored data. This can be advantageous for dynamic datasets where new data points are frequently added. The decision-making process in K-NN is easily interpretable, as predictions are based on nearby instances in the feature space. This transparency can be beneficial in understanding model predictions and explaining outcomes to stakeholders.

Figure 1 represents the block diagram for K-Nearest Neighbor algorithm. The working of K-Nearest Neighbor algorithm is:

1. Data which has been given as input is read and preprocessed.

2. Decision trees are trained on the preprocess data using the Knn.

3. All the decision trees which are trained are combined to prepare a final model.

4. The final output model achieved is used to make the predictions on the dataset.

**Pseudocode**

Step 1. Load the dataset of known individuals, including features like facial features, age, etc.

Step 2. Preprocess the data, ensuring it is in a suitable format for KNN (e.g., scaling features).

Step 3. Input the features of the missing person.

Step 4. Calculate the distance between the features of the missing person and all individuals in

the dataset

Step 5. Sort the distances in ascending order and select the top k nearest neighbors.

Step 6. Determine the majority class among the k nearest neighbors (e.g., using voting).

Step 7. Output the predicted class as the potential identity of the missing person.

**Long short term memory**

LSTMs utilize specialized memory cells that can store information over long periods, preventing the vanishing gradient problem. These memory cells allow LSTMs to capture and remember relevant information from earlier parts of a sequence. It incorporate three gates—input gate, forget gate, and output gate—that regulate the flow of information within the network. These gates enable LSTMs to selectively update and forget information, enhancing their ability to manage long-term dependencies. The input gate determines how much of the new information should be stored in the memory cell. It uses a sigmoid activation function to control the flow of information. They are trained using a variant of the backpropagation algorithm called Backpropagation through Time. This allows the network to learn from sequences of data by adjusting the weights during training. They are widely used in language modeling, sentiment analysis, machine translation, and other NLP tasks due to their ability to capture contextual dependencies in sequential data.

**Pseudocode**

Step 1. Load the dataset of known individuals, including features like facial features, age, etc.

Step 2. Preprocess the data, ensuring it is in a suitable format for LSTM (e.g., scaling features).

Step 3. Input the features of the missing person.

Step 4. Calculate the distance between the features of the missing person and all individuals in

The dataset

Step 5. Sort the distances in ascending order and select the top LSTM.

Step 6. Determine the majority class among the LSTM neighbors (e.g., using voting).

Step 7. Output the predicted class as the potential identity of the missing person.

The present research has been carried out in the system which has hardware specification of Intel i5 as the core processor, RAM of 8 GB, and storage of 512 GB SSD followed by the software specifications which includes Windows 11, Jupyter Notebook, Chrome web browser and SPSS Software for the result analysis. Python programming language with a version of 3.10 has been implemented to obtain the required accuracy for the current research. The program has been executed in the Jupyter Notebook compiler in the current system.

The execution procedures given below,

1. Install the Jupyter Notebook with the help of Anaconda software.

2. A Jupyter notebook can be opened by using the command “jupyter notebook”.

3. Now create a new notebook by clicking on the top right corner.

4. Now write the required python code in the first cell of the notebook.

5. Import some of the required libraries such as numpy, pandas, matplotlib and seaborn.

6. Run the program by clicking the run button.

7. The accuracy should be noted in the excel sheet and run with the help of SPSS software.

The dataset has been collected from the Kaggle website which is a freely available platform which has been used by many of the machine learning and the data scientist’s students for various research purposes. The present dataset is named Finding Missing persons using Ai The Data source link is: (“Kaggle” 2022). The present Dataset consists of 11 columns (attributes) and 10684 rows. The column names in the Dataset are as below:

* Name
* Gender
* Relative
* AgeStart
* AgeEnd
* Height Start
* HeightEnd
* Built
* Dist
* State
* Found

The above dataset has both dependent and independent columns. The independent columns from the above dataset are: Name,Gender,Dist,AgeStart,HeightStart,Built,Dist. The dependent columns which have been collected from the above dataset are: Relative,Found,State,AgeEnd,HeightEnd. The dataset which has been considered here is in the text form and there are no images or audio files included in the complete dataset. The dataset training and testing are planned in the ratio of 80:20 respectively.

**Statistical Analysis**

The statistical analysis for the current research is carried out using the SPSS software of version 26. The values which are collected in excel sheet are inserted into the SPSS software for the analysis of Independent samples T-test among the K-Nearest Neighbor and Longshot term memoryalgorithm.TheindependentvariablesareName,Gender,Dist,AgeStart,HeightStart,Built,Dist. The dependent variables are Relative, Found,State,AgeEnd,and HeightEnd . The Independent samples T-Test analysis has been performed by analyzing the above collected data between the Novel KNN algorithm and LSTM algorithm (S. Ayyappan and S. Matilda 2020).

**RESULTS**

Table 1 shows the T-Test results of the proposed K-Nearest Neighbor algorithm and the Convolutional Neural Network algorithm which has been run numerous times in the Jupyter notebook with a sample size of 20. From Table 1, it has been observed that the accuracy of the algorithm is K-Nearest Neighbor algorithm 87.20% and for the long short term memory algorithm the accuracy is found to be 75.88%. The standard deviation and the Standard Error Mean has also been calculated for the K-Nearest Neighbor algorithm and long short term memory algorithm

Table 2 represents the outcome of the analysis of the Independent samples test which has been performed for the K-Nearest Neighbor algorithm and the Long short term memory algorithm. From Table 2, the significance value for the one tailed test is found to be 0.937, two-tailed is 0.000 and it is found that the Independent samples test has been carried out at Confidence Interval of 95%.

Figure 2 shows the comparison graph between the K-Nearest Neighbor algorithm and the long short term memory algorithm. From the graph, it is concluded that our proposed K-Nearest Neighbor algorithm has an accuracy of 87.20% and the long short term memory algorithm has an accuracy of 75.88%. The plots of the graph are shown in the figure below in Fig. 1

**DISCUSSION**

In this research the K-Nearest Neighbor algorithm is compared with the algorithm to predict the future missing persons using ai and to enhance the accuracy of the existing system. By performing the experiment K-Nearest Neighbor algorithm has achieved an accuracy of 87.20% and long short term memory has achieved an accuracy of 75.88%. The significance value for this research is found to be 0.000 after performing the Independent samples T-test analysis. The K-Nearest Neighbor algorithm (87.20%) has been found to be more accurate than the Long short term memory algorithm (75.88%). The significance value is found to be 0.000 which is lesser than 0.05 (p<0.05), therefore it is observed that the two groups are statistically significant.

In the recent survey, the K-Nearest Neighbor algorithm has been found to have more promising accuracy than the other real world algorithms (GE Xinliang, YANG Jie 2007). The present framework will combine the two datasets with the data collected from the users and found that theK-Nearest Neighbor has provided the best accuracy (Florian Schrof, DmitryKalenichenko 2015).K-Nearest Neighbor algorithm has been significantly faster than the other gradient boosting methods and has more precise accuracy by all means. The results can be developed by implementing new features and picking the best data set (L.Wnag,D.He 1990). Some of the researchers have proposed theK-Nearest Neighbor algorithm in some of their research articles and concluded that the K-Nearest Neighbor algorithm has provided better results than the other Machine Learning algorithms (Y. Lin, L. Zheng, and Z. Zheng 2019). Some of the articles have proposed the Random Forest algorithm to forecast the future charges of finding missing persons using ai and found it had provided better accuracy in some cases than our proposed (S. Abhilash and V. M. Nookala 2022). In some research surveys some of the researchers have implemented the K-Nearest Neighbor (KNN) algorithm to provide future prices of flight tickets to the customers in an efficient way and it has been implemented and found out that it provided more accurate results than our Novel XGBRegression algorithm (CAO Xd, WEI YC 2014).

**CONCLUSION**

The aim of the present experimentation research is to improve the accuracy of finding missing persons using ai and help the customers to book the tickets accordingly. In this research article the K-Nearest Neighbor algorithm is compared with the long short term memory algorithm. Results which had been obtained showed that the K-Nearest Neighbor algorithm has provided an accuracy of 87.20% and long short term memory algorithm has recorded 75.88% of accuracy

**DECLARATIONS**

**Conflict of Interests**

No conflict of Interest in this manuscript.

**Authors Contributions**

Author R.Rupasridevi was involved in data collection, data analysis and manuscript writing. Author C.Clement raj was involved in the conceptualization, data validation and critical review of manuscript.

**Acknowledgements**

The authors would like to express their gratitude towards Saveetha School of Engineering, Saveetha Institute of Medical And Technical Sciences (Formerly known as Saveetha University) for providing the necessary infrastructure to carry out this work successfully.

**Funding**

We thank the following organizations for providing financial support that enabled us to complete the study

1. Mind Play Software Services Pvt. Ltd, Chennai

2. Saveetha School of Engineering

3. Saveetha Institute of Medical And Technical Sciences

4. Saveetha University

**REFERENCES**

Abrah&atildeo, W., Oliveira, G., Silguero, L., Diaz, D.H., Gomez, M.A. and Barbosa, J., A comparison of Haar-like, LBP and HOG approaches to concrete and asphalt runway detection in high resolution imagery

CAO X D, WEI Y C, WEN F, et al. Face alignment by explicit shape regression[J]. International Journal of Computer Vision, 2014, 107(2): 2887-2894.

Chang-Yeon, J., 2008. Face Detection using LBP features. Final Project Report, 77.

Florian Schrof, Dmitry Kalenichenko and James Philbin “FaceNet: A Unified Embedding for Face Recognition and Clustering” arXiv: 1503.03832V3 [cs.CV], 17 June 2015.

GE Xinliang, YANG Jie, LI Feng, et al. Statistical Model-Based Face Pose Estimation [J]. Journal of Tianjin University (English Edition), 2007, 13(2): 152-156.

L.Wang and D. He, Texture classification using texture spectrum, Pattern Recognition, 23(8)905-910,1990.

REN S, CAO X, WEI Y, et al. Face alignment at 3000 fps via regressing local binary features[C]. Computer Vision and Pattem Recognition(CVPR). IEEE, 2014: 1685-1692.

SCHULTER S, LEISTNER C, WOHLHART P, et al. Accurate object detection with joint classification regression random forests [J]. Fuel Processing Technology, 2014, 91(6): 591-599.

Sarabjit Singh, AmritpalKaur, Taqdir, A Face Recognition Technique using Local Binary Pattern Method, International Journal of Advanced Research in Computer and Communication Engineering Vol.4, Issue 3, March 2015.

Z. Xue Mei, W. Cheng Bing “A Real-time Face Recognition System Based on the Improved LBPH Algorithm” IEEE 2nd International Conference on Signal and Image Processing 2017

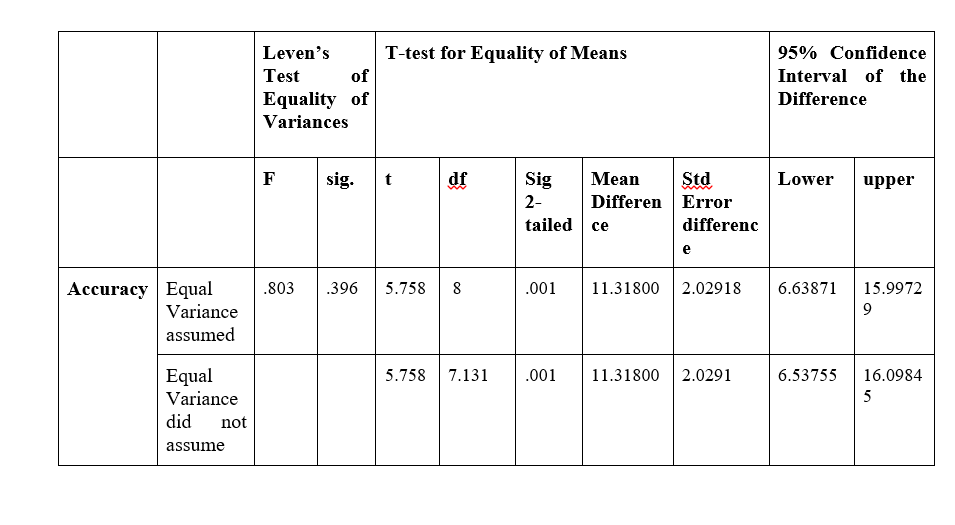
**TABLES AND FIGURES**

Table 1. Presents the statistical analysis results of the K-Nearest Neighbor algorithm and the Convolutional Neural Network algorithm, comparing the mean accuracy, standard deviation, and standard error mean values across 20 sample datasets.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Algorithm** | **N** | **Mean** | **Std.Deviation** | **Std.Error Mean** |
| **Accuracy** | **KNN** | **5** | **87.2000** | **2.58844** | **1.15758** |
| **CNN** | **5** | **75.8820** | **3.72663** | **1.66660** |

Table 2. An independent sample T-Test was conducted to determine the significance of the difference between the two groups, using a significance level of p=0.000 (p<0.05), indicating that the difference is statistically significant.

**Independent Samples Test**



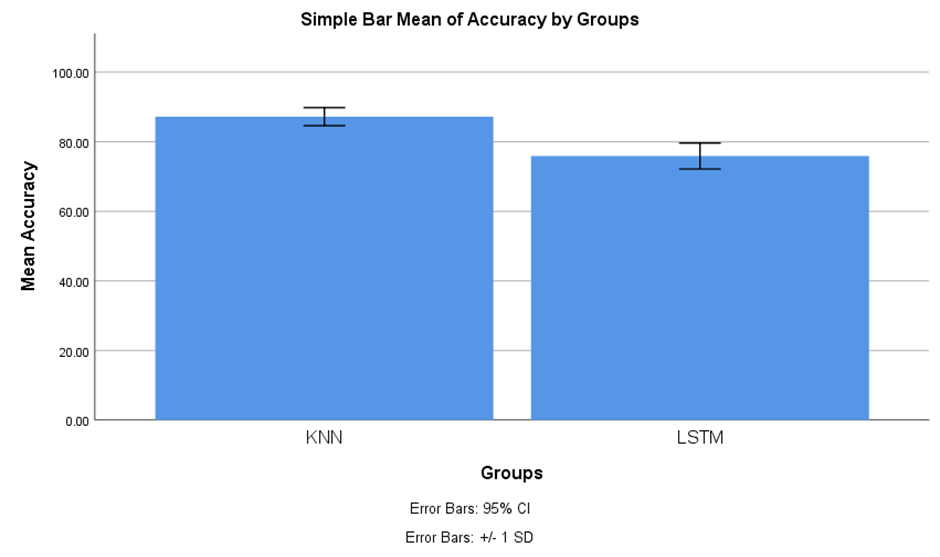


Fig. 1. This figure shows the comparison between the K-Nearest Neighbor algorithm and the Convolutional Neural Network algorithm in terms of Mean Accuracy. The Mean accuracy of the K-Nearest Neighbor is better than the Mean accuracy of the Convolutional Neural Network algorithm. X-axis: K-Nearest Neighbor algorithm vs Convolutional Neural Network algorithm, Y-axis: Mean Accuracy. Error Bar +/-1SD.