*Journal of Science and Technology*
*ISSN: 2456-5660 Volume 9, Issue 1 (JAN -2024)*
*www.jst.org.in* *DOI:https://doi.org/10.46243/jst.2024.v9.i1.pp39-49*

# AI Based Detecting Deception in Online Interactions: An Analysis of the Dishonest Internet Users

A. Sneha[1], U. Leenasri[2], V. Anusha[2], S. Shirisha[2]

[1]Assistant Professor, [2]UG Student, [1,2]Department of Computer Science Engineering
[1,2]Malla Reddy Engineering College for Women, Maisammaguda, Dhulapally, Kompally, Secunderabad-500100, Telangana, India

***To Cite this Article***

A. Sneha[1], U. Leenasri, V. Anusha, S. Shirisha **, "AI Based Detecting Deception in Online Interactions: An Analysis of the Dishonest Internet Users"** *Journal of Science and Technology, Vol. 09, Issue 01 - JAN 2024, pp39-49*

***Article Info***

*Received: 25-12-2023 Revised: 05 -01-2024 Accepted: 15-01-2024 Published: 25-01-2024*

## ABSTRACT

With the widespread adoption of the internet, online interactions have become an integral part of modern communication. However, this surge in digital interactions has also brought about a significant rise in deceptive practices, ranging from misinformation and fraud to identity theft and cyberbullying. Detecting and mitigating these dishonest behaviors has become a critical concern for maintaining trust and integrity in digital communities. The primary challenge lies in developing a robust and automated system capable of identifying deceptive content amidst the vast volume of online interactions. In the absence of advanced AI-based systems, deception detection in online interactions has heavily relied on manual monitoring, keyword-based filters, and rule-based algorithms. These conventional methods are limited in their effectiveness, as they struggle to adapt to evolving deceptive tactics and often generate false positives or negatives. Therefore, the need for effective deception detection systems in online interactions has never been more pressing. The advent of social media, e-commerce, and various online forums has created an environment where deceptive practices can have far-reaching consequences. Ensuring the safety and trustworthiness of these platforms is imperative for user confidence, cybersecurity, and the overall well-being of online communities. Hence, by utilizing machine learning algorithms, advanced linguistic analysis, and behavioral pattern recognition, this research aims to develop a powerful tool capable of accurately discerning deceptive from genuine online interactions. Through the integration of multi-modal approaches and feature engineering, the proposed system promises to significantly enhance the accuracy and efficiency of deception detection in digital communities, ultimately fostering a safer and more trustworthy online environment.

**Keywords:** Dishonest Internet Users, Artificial Intelligence, Detecting deception, Online Interactions.

## 1. INTRODUCTION

The exploration of detecting deception in online interactions stems from the rapid evolution and widespread integration of the internet into modern communication. As online interactions became ubiquitous, so did the emergence of deceptive practices, posing significant threats ranging from misinformation and fraud to identity theft and cyberbullying. The escalating prevalence of these dishonest behaviors has elevated the urgency to develop effective methods for identifying and mitigating them to maintain trust and integrity in digital communities.

Historically, the challenge of deception detection in online interactions was primarily addressed through manual monitoring, keyword-based filters, and rule-based algorithms. However, these conventional methods demonstrated limitations in their adaptability to evolving deceptive tactics, often resulting in either false positives or false negatives. The absence of advanced AI-based systems meant that the effectiveness of online deception detection was hampered, leaving digital platforms vulnerable to deceptive practices.

The rise of social media, e-commerce platforms, and various online forums further exacerbated the challenges, as deceptive practices carried the potential for far-reaching consequences in terms of user confidence, cybersecurity, and the overall well-being of online communities. Recognizing the pressing need for more robust and automated deception detection systems, this research has emerged to address the deficiencies of existing methods.

By leveraging machine learning algorithms, advanced linguistic analysis, and behavioral pattern recognition, this research seeks to pioneer the development of a powerful tool capable of accurately discerning deceptive from genuine online interactions. The integration of multi-modal approaches and feature engineering represents a significant departure from traditional methods, promising to enhance the accuracy and efficiency of deception detection in digital communities. The historical context underscores the transformative potential of this research, aiming to foster a safer and more trustworthy online environment by staying ahead of the evolving landscape of deceptive practices on the internet.

## 2. LITERATURE SURVEY

There has been a long history of human interest in identifying deceptive behaviour. Trovillo (1939) addressed the historic evidence date back to the Hindu Dharmasastra of Gautama (900 – 600 BCE) and the Greek philosopher Diogenes (412 – 323 BCE). In 1921, Larson invented the Polygraph (Larson et al., 1932), which has been considered as one of the popular methods for lie detection and works by measuring physiological changes in a person in accordance with stress factors. Typically, the polygraph instrument captures physiological changes such as pulse rate, blood pressure and respiration that can be interpreted by psychological experts to identify truthful or deceptive behaviour. With respect to different scenarios, a polygraph test takes up to four hours which leads to limitations on it use in real time conditions. Research studies have been supporting the validity of the polygraph as well as criticizing its use in specific cases. A meta-study by Axe et al., (Axe et al., 1985) found 10 studies from a pool of 250 (that were sufficiently rigorous to be included), indicated that the controlled question test could perform significantly better than chance under specified narrow conditions. However, the deception classification contained a high number of false positives, false negatives and inconclusive instances. In addition, substantial information about the interviewee's background (e.g. occupation, work record and criminal record) was required to be captured before the examination in order to construct a good set of control questions.

Vocal cues, voice stress and acoustic features have also been employed as indicators to distinguish the act of deceit (Hirschberg, 2005). Distinctive additional micro tremors appear due to cognitive

overload during the deceptive behaviour (Walczyk, 2013). However, the performance of deception detection using voice stress analysis has been described as "charlatanry" (Eriksson & Lacerda, 2007). Likewise, linguistics has also investigated the changes in language and its structure to classify signs of deception. Linguistic inquiry and word count analysis for deception detection revealed that truth tellers' statements contain more first-person pronouns and self-references (e.g. mine, our) while liars statements contain more words referring to certainty (e.g. totally, truly) and to other- references (they, themselves) (Eriksson & Lacerda, 2007; Abouelenien et al., 2017). A variety of statistical features including mean length of sentence, mean length of clause and clauses per sentence have been extracted from transcribed interviews to evaluate the linguistic hypothesis that liars use less complex and less detailed sentences.

Vrij et al., (Vrij, 2009) reported on the use of thermal imaging of the facial periorbital area to analyse the variations in blood flow specifically when answering unexpected questions. A thermal facial pattern-based approach introduced by (Pavlidis et al., 2002) claims the deception detection accuracy is comparable to that of polygraph tests. Likewise, a thermodynamic model of blood flow variations using the thermal images of facial periorbital area to detect the deceptive behaviour is presented in (Pavlidis and Levine, 2001, Pavlidis et al., 2002). Relationships between different facial emotions (such as stress, fear, and excitement) and deceptive behaviour using thermal imaging is addressed in (Merla and Romani, 2007). Basher and Reyer, 2014) used thermal variation monitoring of the periorbital region and a nearest neighbor classifier that was trained on a high-dimensional feature vector extracted using an average value from each sub-region to detect deception. Experimental results indicated that the classification accuracy did not differ significantly from a random chance distribution based on leave-one-person-out methodology and five-fold cross validation.

In addition to the aforementioned methods, analysis of eye interactions and facial micro-expressions also have been studied as a non-verbal deception detection method (Ekman, 2001). During the act of deceit, relatively short involuntary facial expressions may appear that can be helpful to detect deceptive behaviour. Furthermore, the analysis of facial expressions in terms of asymmetry and smoothness features (Ekman, 2003) indicate their relationship with the deceptive behaviour. Face orientation and intensity of facial expressions is also used to classify the act of deceit (Tian et al., 2005). Likewise, geometric features (Owayjan, et al., 2012) and micro-expressions (Pfister and Pietikäinen, 2012) extracted from the facial data have also been used to classify the deceptive behaviour. Related research in (Pons and Masip, 2018) indicated the usefulness of facial micro-gestures towards the identification of comprehension levels. Buckingham et al., (2014) used artificial neural networks sequentially to identify the micro-gestures and perform the classification respectively. Pérez-Rosas et al., (Rosas et al., 2015) proposed the multi-model deception detection methodology that used a novel dataset acquired from real public court trials. A variety of linguistic and gesture modalities including facial features were combined together to classify the deceptive behaviour. Results reported a classification accuracy between 65 and 75% with varying combinations of modalities. Furthermore, the results indicated that the system outperformed human experts in terms of correct identification of deceptive behaviour. One of the recent machine-based research studies that uses the direction of gaze, eye movements and blink rate to distinguish the truthful and deceptive behaviours is presented in (Borza et al., 2018). The research outcomes indicated the normalised eye blink rate was an important clue of deception detection. Research carried out in (Marchak, 2013, Nunamaker et al., 2016, Levine, 2014, Schuetzler, 2012, Kumar, 2016, Pak and Zhou, 2011, Lim et al., 2013) also indicate the significance of eye interaction and associated corresponding features towards effective deception detection. Eyes blink rate, pupil dilation and gaze are the most

common examples of such a feature set. Research studies indicate the relationship between these attributes and cognitive effort variations in deceptive and truthful subjects (Fukuda, 2001). Like other psychological clues for deception detection, additional cognitive efforts performed by deceivers undergo additional cognitive processes compared to truthful individuals that leads to an increased pupil diameter for deceivers (Proudfoot et al., 2015, Dionisio et al., 2001). In a similar study by Marchak (Marchak, 2013), compared to truthful participants, a suppressed eye blinking rate is noticed for participants involved in a mock crime to transport an explosive device to be used for a disturbance.

## 3. PROPOSED SYSTEM

### Overview

In response to these challenges. The essence of the AI-driven approach involves training these models on meticulously labeled datasets containing examples of different classes. Through this training process, the models can autonomously learn to extract relevant features from internet users dataset, enabling to discern and classify classes or labels with heightened accuracy.

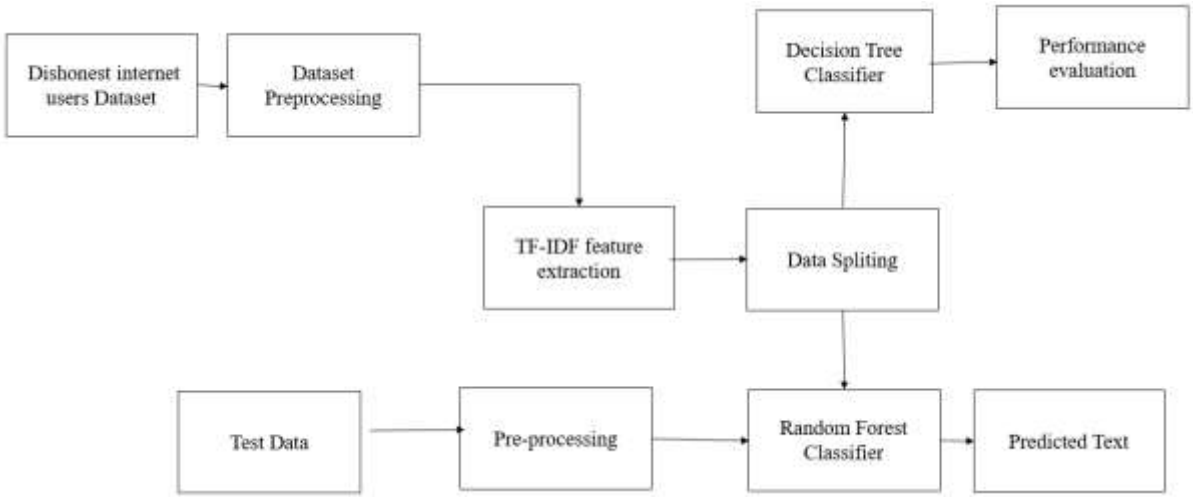


Figure.1: Architectural block diagram of proposed system.

The provided Python script implements a graphical user interface (GUI) application using Tkinter for a surface identification project based on robot-sensed data. Here's a detailed explanation of the steps carried out by the application:

Dataset Upload: The application starts with a button labeled "Upload Dataset." When clicked, this button opens a file dialog, allowing the user to select the dataset file (assumed to be in CSV format). The chosen file is then loaded into the application, and its name is displayed in the text widget. The dataset is stored in the 'dataset' variable.

Dataset Preprocessing: The "Preprocess Dataset" button triggers the preprocessing phase. Missing values in the dataset are filled with zeros, and an overview of the dataset, including the first few records, is displayed in the text widget. Additionally, a count plot is generated to visualize the distribution of classes in the ‘label’ column. Label encoding is applied to convert categorical class labels into numerical values.

Train-Test Splitting: The dataset is split into training and testing sets using the scikit-learn train_test_split function. Information about the total number of records in the dataset, as well as the training and testing sets, is displayed in the text widget.

Decision Tree Classifier: The "Decision Tree Classifier" button initiates the training of a Decision Tree classifier. The model is fitted on the training set, and predictions are made on the testing set. The evaluation metrics, including accuracy, confusion matrix, and classification report, are displayed. Additionally, a Receiver Operating Characteristic (ROC) graph is generated to visualize the model's performance.

Random Forest Classifier: The "Random Forest Classifier" button triggers the training of a Random Forest classifier. Similar to the Decision Tree model, evaluation metrics and a ROC graph are displayed in the text widget.

Prediction on Test Data: The "Prediction" button allows the user to select a file for making predictions using the trained Decision Tree classifier. Predictions are displayed in the text widget, indicating the predicted classes for each test data entry.

Performance Estimation and Comparison: The "Comparison Graph" button generates a bar graph comparing performance metrics (precision, recall, F1-score, and accuracy) between the Decision Tree classifier and the Random Forest classifier. This visual representation provides an easy comparison of the two models.

Exit: The "Exit" button closes the Tkinter GUI application.

**Random Forest Algorithm**

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

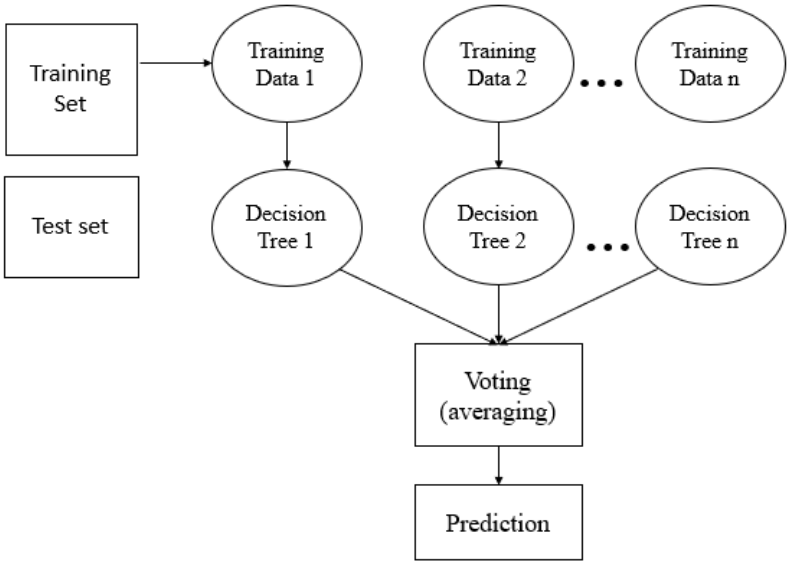


Fig.2: Random Forest algorithm.

**Random Forest algorithm**

Step 1: In Random Forest n number of random records are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

**Advantages**

- The presented Tkinter-based surface identification project utilizing Decision Tree and Random Forest classifiers offers several advantages:
- User-Friendly Interface: The graphical user interface (GUI) created with Tkinter enhances user interaction by providing buttons for various functionalities. This makes the application accessible and easy to use for individuals without programming expertise.
- Dynamic Dataset Upload: The ability to upload datasets through the "Upload Dataset" button allows users to work with diverse datasets effortlessly. This dynamic approach supports the application's adaptability to different use cases and datasets.
- Comprehensive Preprocessing: The "Preprocess Dataset" button automates preprocessing steps, such as handling missing values and label encoding. The generated count plot aids in visualizing the distribution of classes, offering insights into the dataset's characteristics.
- Transparent Train-Test Splitting: The application transparently communicates the process of splitting the dataset into training and testing sets. Information about the total records and the sizes of the training and testing sets is provided, enhancing transparency in the data preparation phase.
- Multiple Classifier Options: The inclusion of both Decision Tree and Random Forest classifiers offers flexibility to users. They can choose between different algorithms based on

the nature of their data and the problem at hand, allowing for experimentation and model comparison.

- Performance Metrics and Visualization: The application computes and displays essential performance metrics, including accuracy, confusion matrix, and classification report. The incorporation of ROC curves visually represents the models' performance, aiding users in assessing the classifiers' ability to discriminate between classes.
- Prediction on Test Data: The "Prediction" button allows users to make predictions on new test data using the trained Decision Tree classifier. This functionality is valuable for real-world applications where the model is deployed on unseen data.
- Comparison Graph: The "Comparison Graph" button generates a bar graph comparing performance metrics between the Decision Tree and Random Forest classifiers. This visual representation facilitates a quick and clear understanding of how different algorithms perform on the given dataset.
- Scalability and Adaptability: The modular structure of the application makes it scalable and adaptable. Users can extend the functionality by adding more classifiers or incorporating additional preprocessing steps to suit specific project requirements.

## 4. RESULTS AND DISCUSSION

**Implementation description:**

The Python code that uses the Tkinter library to create a graphical user interface (GUI) application for detecting dishonest internet users. The application employs Natural Language Processing (NLP) techniques, such as text preprocessing and machine learning models (Random Forest Classifier and Decision Tree Classifier), to analyze and classify online interactions as either "honest" or "dishonest." Here's a brief overview of the main components and functionalities:GUI Components: Labels, buttons (Upload Dataset, Preprocess and Split Data, TF-IDF Feature extraction, Dataset Splitting, Train Random Forest Model, Train Decision Tree Model, Predict Text), a text widget, and a scrollbar are defined.

- Button actions are associated with specific functions.
- Functionality:
- uploadDataset: Opens a file dialog to select the dataset file (CSV format) and loads it into the application.
- preprocessText: Performs text preprocessing on each text in the dataset, including converting to lowercase, removing special characters, numbers, punctuation, and stopwords.
- extractTfidfFeatures: Uses TF-IDF vectorization to convert the preprocessed texts into numerical features.
- splitData: Splits the TF-IDF features and labels into training and validation sets.
- preprocessAndSplit: Preprocesses the loaded dataset and displays the results.
- TF_IDF: Performs TF-IDF feature extraction and displays the results.
- Data_split: Splits the dataset into training and testing sets and displays the sizes.
- performance_evaluation: Evaluates the performance of a machine learning model (accuracy, precision, recall, F1-score, confusion matrix) and displays the results.
- trainRandomForestModel: Trains a Random Forest Classifier model using the TF-IDF features and evaluates its performance.

**Dataset description:**

- text: This column contains textual data of comments or paragraphs written by individuals. The content varies, including expressions of appreciation, discussions about personal experiences, and biographical information about a person or an entity.
- Label: This column represents labels or categories associated with each text entry. The label is binary, with 0 indicating a certain category or sentiment (potentially non-controversial or neutral content) and 1 indicating another category or sentiment (potentially controversial or negative content).
- The dataset is a collection of text entries, sourced from online comments or other textual sources, labeled with a binary classification indicating the nature of the content. The goal of the dataset is sentiment analysis, content moderation, or another task where the text needs to be categorized based on its content.

**Results description:**

This figure 3 represents the graphical user interface (GUI) of the application for detecting dishonest internet users. It has buttons and options for various functionalities.

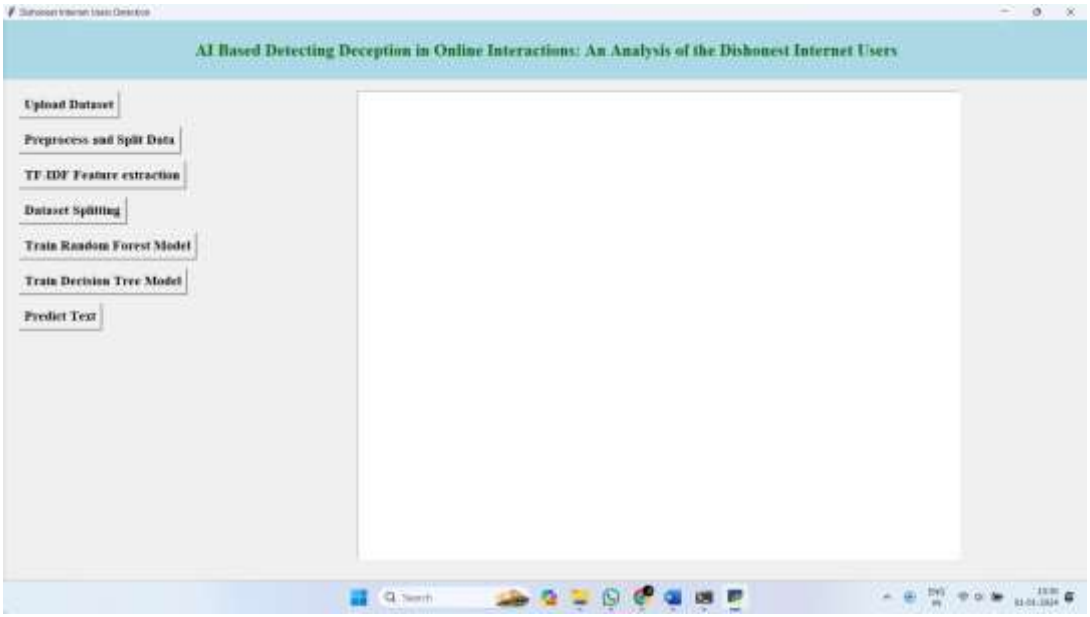


Figure 3: Presents the GUI of dishonest internet users.

The figure 4 below shows the interface or message confirming the successful loading of the dataset after the user uploads a CSV file.


Figure 4: Displays the loaded uploaded dataset.

The figure 5 provide confirming the completion of the dataset splitting into training and testing sets.

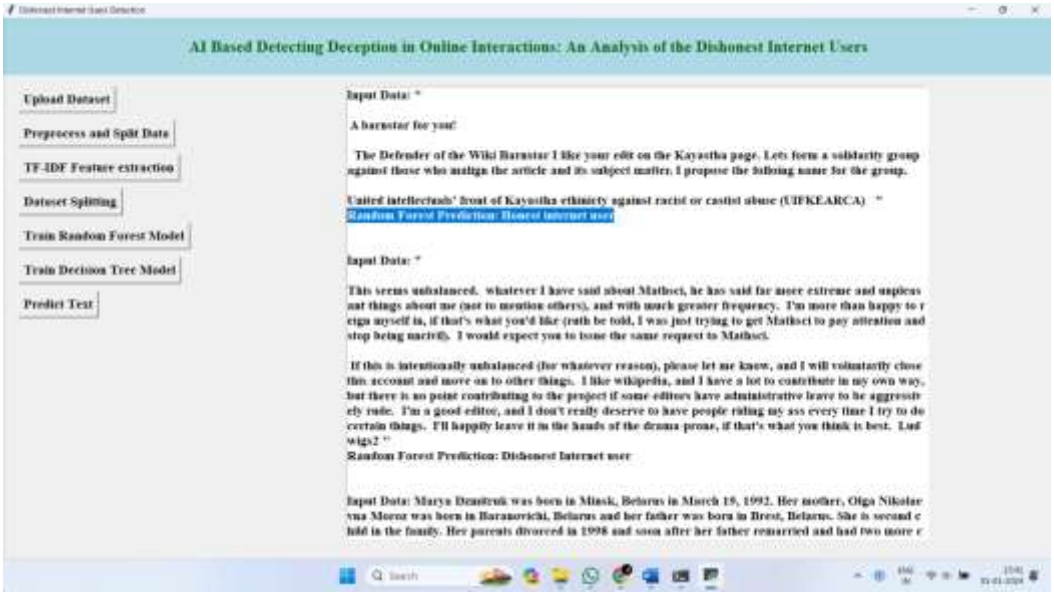


Figure 5: Displays the model predicted outcome label for the test data.

## 5. CONCLUSION AND FUTURE SCOPE

The increasing prevalence of deceptive practices in online interactions necessitates advanced and automated systems to effectively detect and mitigate dishonest behaviors. Traditional methods, relying on manual monitoring and rule-based algorithms, fall short in adapting to the dynamic nature of deceptive tactics in the digital realm. This research addresses this critical challenge by proposing a sophisticated AI-based system for detecting deception in online interactions. The utilization of machine learning algorithms, advanced linguistic analysis, and behavioral pattern recognition represents a significant advancement in the field of deception detection. By integrating multi-modal approaches and feature engineering, the proposed system aims to enhance accuracy and efficiency. This is crucial for maintaining trust, integrity, and user confidence in the digital communities that

have become integral parts of our daily lives. The research not only acknowledges the urgency of the issue but also proposes a solution that aligns with the technological landscape of modern communication. The importance of fostering a safer and more trustworthy online environment cannot be overstated, considering the far-reaching consequences of deceptive practices on social media, e-commerce, and various online forums

**REFERENCES**

1. Abouelenien et al., 2017 M. Abouelenien, V. Pérez-Rosas, R. Mihalcea, M. Burzo Detecting deceptive behavior via integration of discriminative features from multiple modalitiesIEEE Transactions on Information Forensics and Security, 12 (05) (2017), pp. 1042-1055, 10.1109/TIFS.2016.2639344

2. Abouelenien et al., 2014 Abouelenien, M., Rosas, V. P., Mihalcea, R., & Burzo, M. (2014). Deception detection using a multimodal approach. 16th International Conference on Multimodal Interaction (ICMI '14). ACM, New York, NY, USA, 58-65, doi: https://doi.org/10.1145/2663204.2663229.

3. Aristoklis et al., 2005A.D. Anastasiadis, G.D. Magoulas, M.N. VrahatisNew globally convergent training scheme based on the resilient propagation algorithm Neurocomputing, 64 (2005), pp. 253-270, 10.1016/j.neucom.2004.11.016

4. Axe et al., 1985 L. Axe, D. Dougherty, T. Cross The validity of polygraph testing: Scientific analysis and public controversy American Psychologist, 40 (03) (1985), pp. 355-366, 10.1037/0003-066X.40.3.355

5. Basher and Reyer, 2014 Bashar, A., & Reyer, Z. (2014). Thermal Facial Analysis for Deception Detection. IEEE Transactions on Information Forensics and Security. 09(06), 1015-1023, doi: 10.1109/TIFS.2014.2317309.

6. Bond and DePaulo, 2006 C.F. Bond Jr., B.M. DePaulo Accuracy of Deception Judgments Pers Soc Psychol Rev, 10 (3) (2006), pp. 214-234, 10.1207/s15327957pspr1003_2

7. Borza et al., 2018 D. Borza, R. Itu, R. Danescu In the Eye of the Deceiver: Analyzing Eye Movements as a Cue to DeceptionJournal of Imaging, MDPI, 4 (10) (2018), pp. 1-20, 10.3390/jimaging4100120

8. Bradski, 2000 G. Bradski OpenCV Library Retrieved from https://docs.opencv.org/master/d2/d42/tutorial_face_landmark_detection_in_an_image.html

9. Breiman, 2001 L. Breiman Random forests Machine learning, 45 (01) (2001), pp. 5-32, 10.1023/A:1010933404324

10. Buckingham et al., 2014 F. Buckingham, K. Crockett, Z. Bandar, J. O'Shea FATHOM: A Neural Network-based Non-verbal Human Comprehension Detection System for Learning Environments IEEE SSCI, Florida, 403–409 (2014), 10.1109/CIDM.2014.7008696

11. Chen et al., 2018 J. Chen, Z. Chen, Z. Chi, H. Fu Facial expression recognition in video with multiple feature fusion IEEE Transactions on Affective Computing, 9 (1) (2018), pp. 38-50, 10.1109/TAFFC.2016.2593719

12. Cortes and Vapnik, 1995 C. Cortes, V. Vapnik Support-vector networksMachine Learning, 20 (3) (1995), pp. 273-297, 10.1007/BF00994018

.