iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VI **Month of publication:** June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72827

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

Symptoms based Disease Prediction Web Application

Nabarun Ray[1], Soham Das[2], Shuvra Naskar[3], Suvradeep Maity[4]
[1, 2, 4]*dept. of computer science and engineering (IOTCSBT)*
[3]*dept. of electronics and communication engineering Institute of Engineering and Management Kolkata, India*

Abstract: ***Nowadays the way people seek health-related information that is undergoing significant changes worldwide. One major challenge is the difficulty that individuals face in finding trustworthy online resources or information about illnesses, diagnoses, and treatment options. A solution lies in creating a recommendation system for doctors, patients and medications that utilizes review mining to simplify this process and save time. Four machine learning algorithms are used together to achieve around 96% accuracy. It provides departments and doctors information based on prediction to users. It focuses on securely storing patient history for future references integrating advanced password hashing techniques as well as JWT (Jason Web Token) based authentication. It aims to improve healthcare accessibility to reduce patient confusion for faster as well as informed decision making. However, understanding complex medical language can be overwhelming for non-specialist users. The vast amount of medical data is there across different platforms which further complicates their search. To address these issues, such a system must be designed to cater specifically to the needs of the healthcare sector, ensuring it remains user-friendly and easy to understand for all individuals.***

Keywords: Health sector, Disease's diagnosis, securing patient history, Symptoms based diseases prediction.

I. INTRODUCTION

Nowadays, everyone is turning to the internet to seek information regarding diseases, diagnoses, and treatments. They wish to aware themselves with relevant healthcare topics. However, the large amounts of data spread across the internet are difficult to manage, as the information is both vast and advanced. With little medical knowledge, laypeople often struggle with the diagnosis themselves because of the complicated medical jargons. To make things worse, there is no single comprehensive guide or repository which puts everything together in an orderly manner, making navigation difficult and wasting one's time in understanding the basics first.

There is no recommended system that provides users with personalized recommendations for doctors and medications through review mining techniques. Such a system should not only be able to assist with the understanding of complex medical information but also integrate with other features of the healthcare system, while being user-friendly for all types of users regardless of their medical education level. Solving these interface design problems will make accessing health information more effective and accurate and assist users in navigation and retrieval of health information.

This project's goal is to design an interactive intelligent secure web application that can predict potential diseases from a given user symptom list. The platform will provide guide medical assistance aid to users empowering them medically navigate their health care with preliminary guidance.

The goals include the following:

1) Gather a reliable dataset containing diseases and their corresponding symptoms, then prepare the data for model training.
2) Applying different model algorithms such as Decision Tree, Random Forest, K-Nearest Neighbours (KNN), and Gaussian Naive Bayes to perform disease predictions and evaluate their prediction accuracies.
3) Develop a secure, responsive, and a user-friendly web application that allows for easy input of symptoms and provides automatically controlled data access.
4) Create a systematic user management system that allows for easy storing and accessing prior user prediction records which is useful for potential future research.
5) Mention appropriate medical specialists or doctors to users for timely professional medical intervention based on predicted diseases.
6) Combine outputs of multiple models to improve prediction accuracy and reduce errors.

With these goals, it's possible to provide a system that aids in rapid access to healthcare services while giving automated disease prediction assistance.

II. RELATED WORK

One of the facets of modern healthcare is machine learning, which offers sophisticated solutions for medical diagnosis, designing prognosis and treatment plans [1]. Decision Trees lays down the routes of the decision making, while Random Forests increases accuracy through ensemble learning. KNN thrives with new data, and Naive Bayes offers speedy and simple classification based on the assumed probability. Application of these models in healthcare is centred on the ability to predict diseases formed from patterns of symptoms provided by patients.

KNN is preferred when closeness between patient cases is crucial, for example in the case of flu-like symptoms. Naive Bayes has proved to be effective with medical records even though it uses independence assumption because symptoms happen probabilistically [4]. Ensemble methods which combine more than one model have been shown to improve accuracy and reduce bias in comparison to using a single model [5].

Moreover, deep learning techniques like convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been successfully applied to image and time-ordered health records, particularly in radiology and imaging-based fields, dermatology as well as chronic condition surveillance [6].

The symptom checking services provided by platforms such as WebMD and Ada Health propose possible conditions, but do not provide personalization or recommend doctors. In addition, many do not justify how the forecasts are made eliminating trust from the users. Most research concentrates on single diseases rather than predicting multiple diseases [2] from a symptoms analysis. These gaps are closed with our proposed system that provides personalized prediction structure, specialized guidelines, and a protected design which puts the user first.

III. PROPOSED METHODOLOGY

A. Data processing and Modelling

The whole train dataset has 5945 numbers of records with 163 numbers of columns which represent symptoms. Some symptoms are not giving any impact on the prognosis column which is our target column. So, I need to extract only those columns or symptoms which is impacting the prognosis.This is the extracted symptoms list with 124 symptoms. I have used that list of symptoms to train models.There is total 61 diseases in the prognosis column. It is a classification problem. So, I have assigned numbers from 0 to 60 to all individual diseases. After that I have implemented same pre-processing steps for the test dataset and trained four models which are Decision tree model, Random Forest model, KNN model and Gaussian naïve bayes model.

```
In [202]: print(symptoms_list)
['back_pain', 'constipation', 'abdominal_pain', 'diarrhoea', 'mild_fever', 'yellow_urine', 'yellowing_of_eyes', 'acute_liver_fa
ilure', 'swelling_of_stomach', 'swelled_lymph_nodes', 'malaise', 'blurred_and_distorted_vision', 'phlegm', 'throat_irritation',
'redness_of_eyes', 'sinus_pressure', 'runny_nose', 'congestion', 'chest_pain', 'weakness_in_limbs', 'fast_heart_rate', 'pain_du
ring_bowel_movements', 'pain_in_anal_region', 'bloody_stool', 'irritation_in_anus', 'neck_pain', 'dizziness', 'cramps', 'bruisi
ng', 'obesity', 'swollen_legs', 'swollen_blood_vessels', 'puffy_face_and_eyes', 'enlarged_thyroid', 'brittle_nails', 'swollen_e
xtremeties', 'excessive_hunger', 'extra_marital_contacts', 'drying_and_tingling_lips', 'slurred_speech', 'knee_pain', 'hip_join
t_pain', 'muscle_weakness', 'stiff_neck', 'swelling_joints', 'movement_stiffness', 'spinning_movements', 'loss_of_balance', 'un
steadiness', 'weakness_of_one_body_side', 'loss_of_smell', 'bladder_discomfort', 'foul_smell_of urine', 'continuous_feel_of_uri
ne', 'passage_of_gases', 'internal_itching', 'toxic_look_(typhos)', 'depression', 'irritability', 'muscle_pain', 'altered_senso
rium', 'red_spots_over_body', 'belly_pain', 'abnormal_menstruation', 'dischromic _patches', 'watering_from_eyes', 'increased_ap
petite', 'polyuria', 'family_history', 'mucoid_sputum', 'rusty_sputum', 'lack_of_concentration', 'visual_disturbances', 'receiv
ing_blood_transfusion', 'receiving_unsterile_injections', 'coma', 'stomach_bleeding', 'distention_of_abdomen', 'history_of_alco
hol_consumption', 'fluid_overload', 'blood_in_sputum', 'prominent_veins_on_calf', 'palpitations', 'painful_walking', 'pus_fille
d_pimples', 'blackheads', 'scurring', 'skin_peeling', 'silver_like_dusting', 'small_dents_in_nails', 'inflammatory_nails', 'bli
ster', 'red_sore_around_nose', 'yellow_crust_ooze']

In [203]: print(len(symptoms_list))
94
```

Figure 1: All symptoms

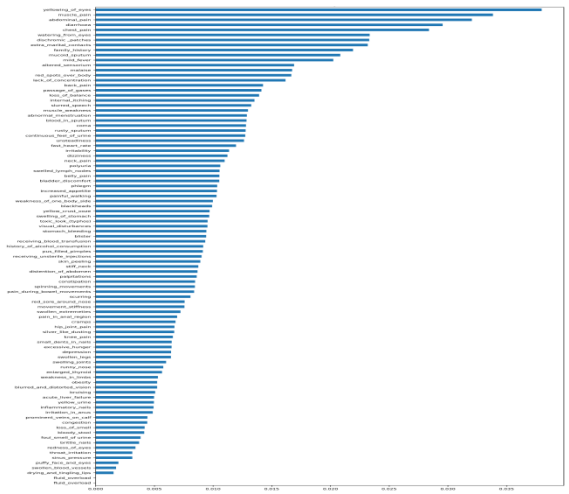


Figure2: Important features

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

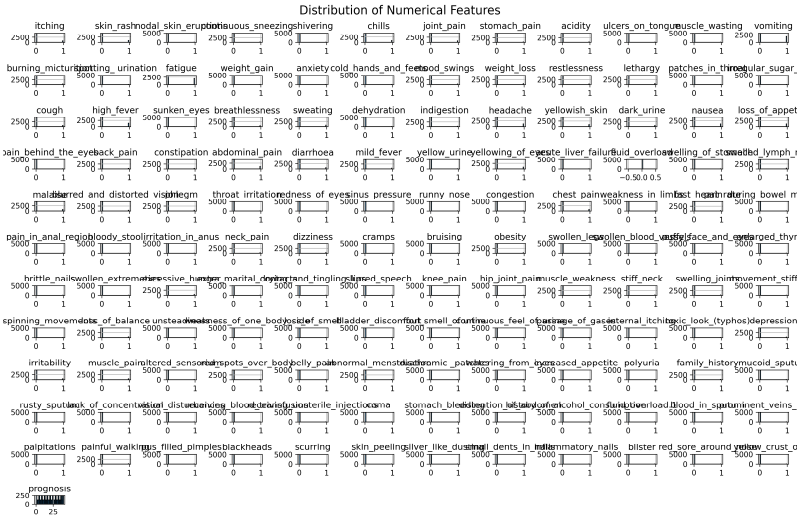


Figure3: Numerical features distribution

It is observed that every model gives accuracy score around 95%, precision score around 92%, recall score around 95% and f1 score around 93%. Here four models are used to avoid any kind of miss prediction when patient is getting information regarding departments and doctors to whom he or she should contact. Usually, any patient fill symptoms as per his or her feeling that sometimes miss interprets the prediction of the diseases. In real life scenario, predictions of four models will cover all cases which can lead to miss prediction. That's how patient can have maximum four departments' doctors' choices in the worst case. On average, patient will get one or two departments' doctors' information. That can speed up the treatment process.

```
[51]: from sklearn import tree
      from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

      # Train the model
      clf3 = tree.DecisionTreeClassifier()
      clf3.fit(X, y)

      # Predict and evaluate
      y_pred = clf3.predict(X_test)
      print("Decision Tree\nAccuracy:")
      print(f"Normalized: {accuracy_score(y_test, y_pred)}")
      print(f"Count: {accuracy_score(y_test, y_pred, normalize=False)}")
      print("Confusion Matrix:")
      print(confusion_matrix(y_test, y_pred))

      # Predict disease from symptoms
      symptoms = ['abdominal_pain', 'diarrhoea', 'mild_fever', 'yellow_urine', 'cramps']
      input_vector = [1 if symptom in symptoms else 0 for symptom in symptoms_list]

      predicted = clf3.predict([input_vector])[0]
```

Figure4: Decision tree model

```
[52]: from sklearn.ensemble import RandomForestClassifier
      from sklearn.metrics import accuracy_score, confusion_matrix

      # Train the model
      clf4 = RandomForestClassifier(n_estimators=100)
      clf4.fit(X, np.ravel(y))

      # Evaluate the model
      y_pred = clf4.predict(X_test)
      print("Random Forest\nAccuracy:")
      print(f"Normalized: {accuracy_score(y_test, y_pred)}")
      print(f"Count: {accuracy_score(y_test, y_pred, normalize=False)}")
      print("Confusion Matrix:")
      print(confusion_matrix(y_test, y_pred))

      # Predict from symptoms
      symptoms = ['abdominal_pain', 'diarrhoea', 'mild_fever', 'yellow_urine', 'cramps']
      input_vector = [1 if symptom in symptoms else 0 for symptom in symptoms_list]

      predicted = clf4.predict([input_vector])[0]
      result[1] = predicted

      if 0 <= predicted < len(disease):
          print(disease[predicted])
```

Figure5: Random forest classifier model

```
[53]: from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix

# Train the KNN model
knn = KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2)
knn.fit(X, np.ravel(y))

# Evaluate the model
y_pred = knn.predict(X_test)
print("kNearest Neighbour\nAccuracy:")
print(f"Normalized: {accuracy_score(y_test, y_pred)}")
print(f"Count: {accuracy_score(y_test, y_pred, normalize=False)}")
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))

# Predict disease based on symptoms
symptoms = ['abdominal_pain', 'diarrhoea', 'mild_fever', 'yellow_urine', 'cramps']
input_vector = [1 if symptom in symptoms else 0 for symptom in symptoms_list]

predicted = knn.predict([input_vector])[0]
result[2] = predicted

if 0 <= predicted < len(disease):
    print(disease[predicted])
```

Figure6: KNN model

```
[54]: from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, confusion_matrix

# Train the Naive Bayes model
gnb = GaussianNB()
gnb.fit(X, np.ravel(y))

# Evaluate the model
y_pred = gnb.predict(X_test)
print("Naive Bayes\nAccuracy:")
print(f"Normalized: {accuracy_score(y_test, y_pred)}")
print(f"Count: {accuracy_score(y_test, y_pred, normalize=False)}")
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))

# Predict disease from symptoms
symptoms = ['abdominal_pain', 'diarrhoea', 'mild_fever', 'yellow_urine', 'cramps']
input_vector = [1 if symptom in symptoms else 0 for symptom in symptoms_list]

predicted = gnb.predict([input_vector])[0]
result[3] = predicted

if 0 <= predicted < len(disease):
    print(disease[predicted])
```

Figure7: Gaussian naïve Bayes model

B. Web application

The web application is created with user interaction and security in mind. It is designed by combining front end and back end together [3]. On the other hand, front end and back end are developed separately.There are Login page, Register page, Prediction page. Patient can login or register using email id & password. After login, patient can fill the form where five symptoms must be filled to get the prediction.

Here in the example, patient is getting the name of the predicted diseases with departments and doctors' information but in real scenario, patient will only get the information regarding departments and doctors. Predicted history will be saved at database with patient details, all filled symptoms, departments and doctors' names.

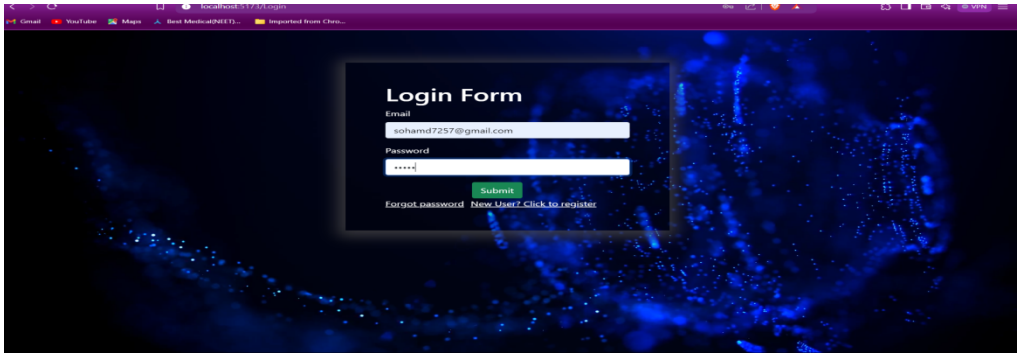


Figure8: Patient login page

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

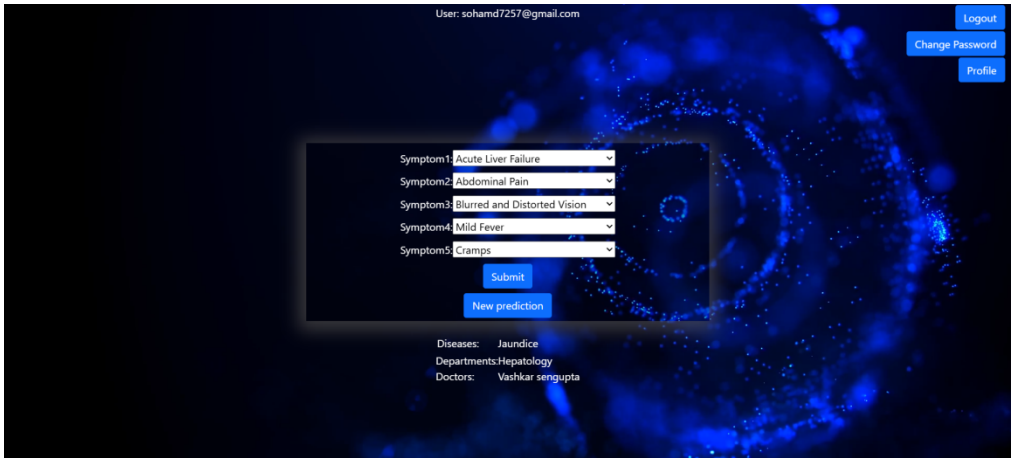


Figure9: Diagnosisresult

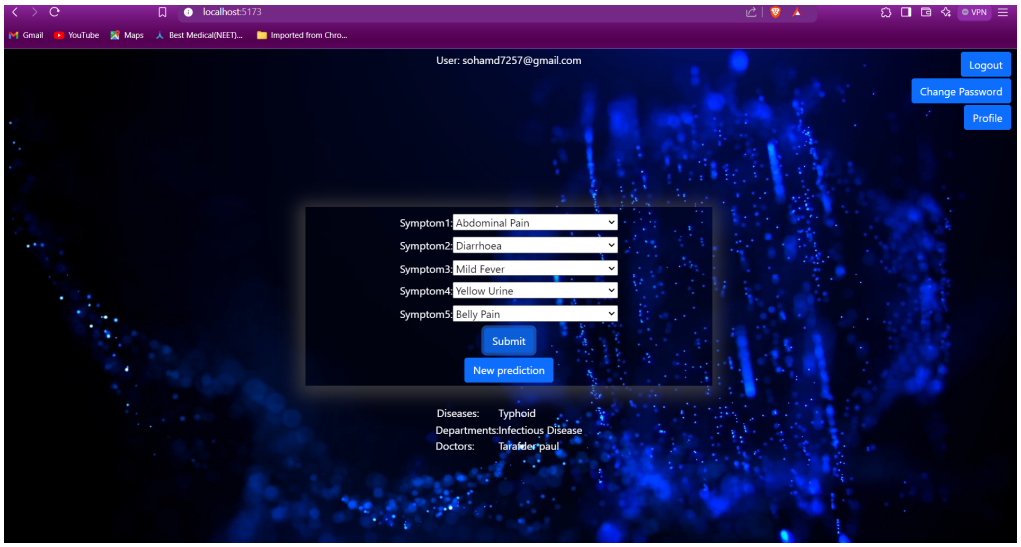


Figure10: Diagnosis result

On right side, the folder structure of the backend is given.On the backend, I made two applications to manage all APIs regarding user account management and diseases prediction.

I have use JWT (JSON Web Token) to manage sessions, reset password both for after and before login. User gets password reset link to registered email id. All passwords of individual user are saved in the database in a hashed form. I have created proctored routes which allow user to surf pages as a logged in user. Without login, user can only access login & register pages.

In the Django admin panel, admin can access all history created by all patients. Those histories can be used for future studies.

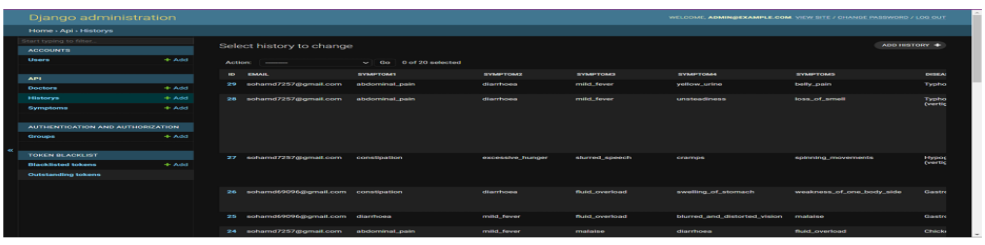


Figure11: Admin panel and Backend structure

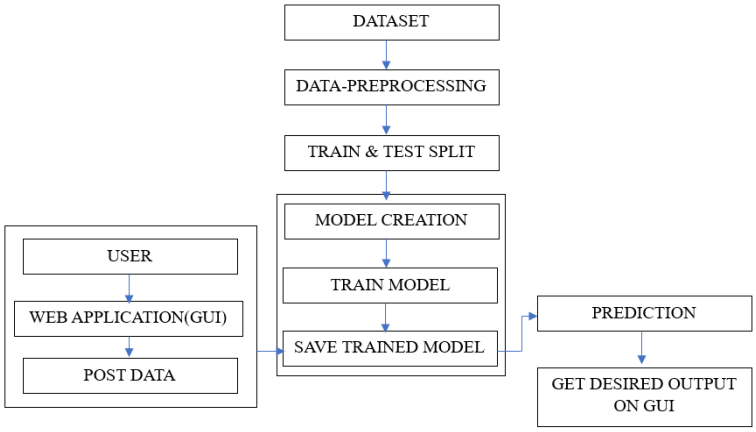


Figure12: Workflow diagram

IV. DISCUSSIONAND CONCLUSION

The implementation of the Symptoms Based Disease Prediction Web Application showcases the potential impacts that machine learning can have in the real world, especially in the healthcare industry. One milestone of the work was constructing a multi-model architecture aimed at predicting diseases with high accuracy based on symptoms provided by users. The performance metrics of the four models published confirm the accuracy and efficiencymachine learning offers for predictive endeavours.In any case, although the technical performance was provided a lot of hope, the greatest takeaway from this project, at least for me, is how it can be applied in the real world. Patients have a hard time of determining which symptoms warrant attention and which doctor or unit they want to see. Our system fills this gap by providing a diagnostic system which will provide department suggestion that stands to significantly reduce the time taken to receive critical care. Some of the salient points learned from the entire endeavours are as follows:

- Quality of data matters. The fact that cleaning the data set and choosing relevant features improves models is enough to prove data quality.
- Using more than one model is useful. It creates redundancy, which, on a broader scope, increases accuracy. In this case, there is decreased instance of incorrect prediction that is crucial in health care.
- User experience is just as important as accuracy. We focused on providing sane interfaces and functional logins while keeping usability in mind for non-technical users.

This project is designed, developed, and deployed as a functioning web-based system that can predict probable diseases using multiple machine learning models and allows the user to enter their symptoms. The results obtained were predictive and justified the approach taken during development. The web application enables users to interact with the system and easily input symptoms. Machine learning model provides predictions on what diseases the user may have given. Recommendations are on which departments and specialists the user should visit.Moreover, the system ensures secure storage of the prediction history, and authentication is protected using JWT tokens, and passwords are maintained in hashed form for privacy protecting data confidentiality and integrity. The application does not attempt to replace medical diagnosis; however, it tries to facilitate in aiding the user seek professional consultation as a preliminary advisory tool. Incorporation into actual healthcare systems – such as a Telemedicine interface or a hospital triage system – could serve as a convincing next step.

V. FUTURE SCOPE

With this work, there are many scopes where improvement work can be done to deliver more valuable service towards end users.

- With a large dataset with more diverse and corelated information, more complex deep learning model can be trained to achieve higher prediction accuracy for complex and rare conditions [4].
- A feature can be implemented, which will suggest next symptoms to the user according to the already selected symptoms. That will surely enhance user experience as well as will increase overall prediction accuracy.

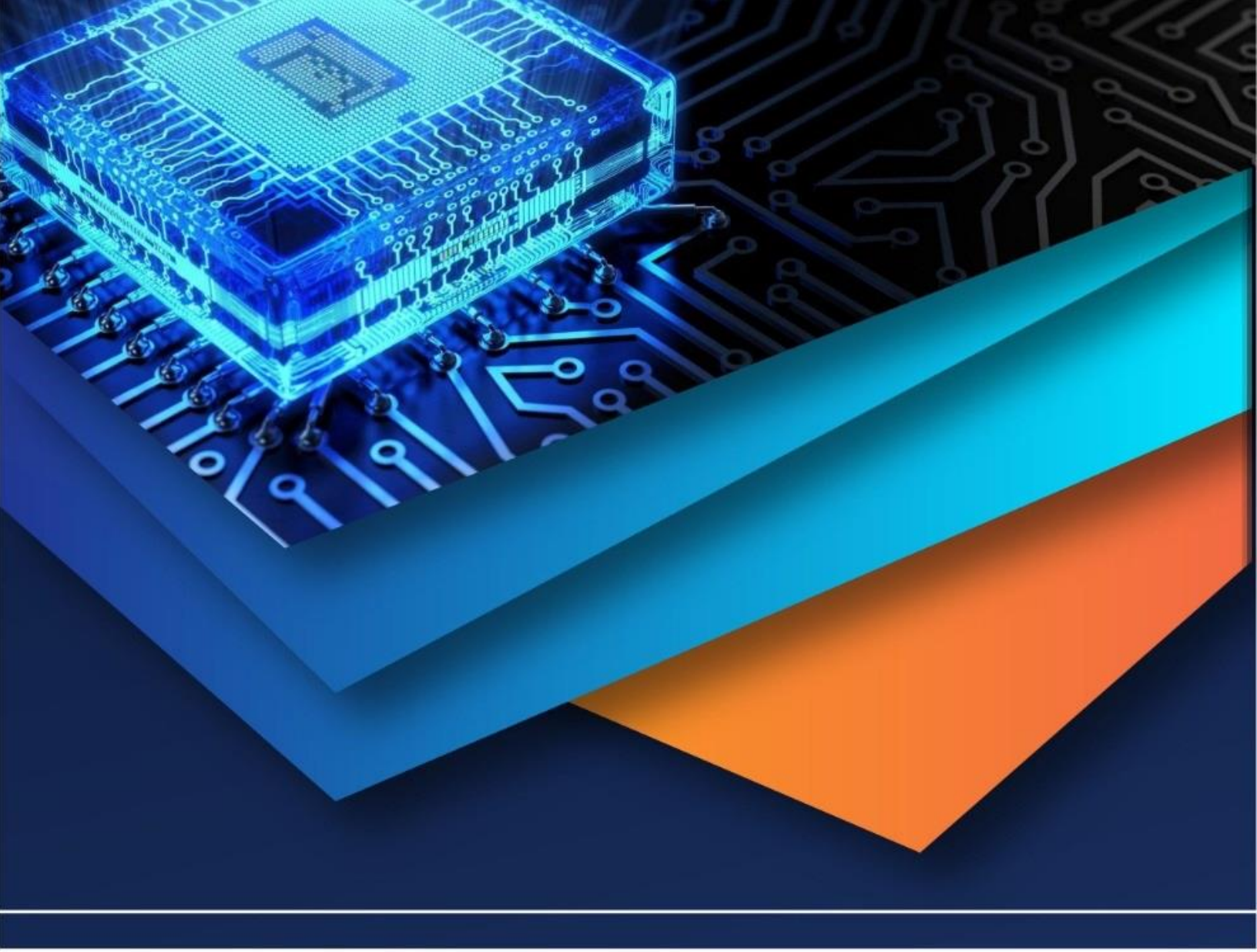
- With the integration of IOT devices, real time health monitoring feature can be enabled which will increase overall prediction accuracy.

VI. ACKNOWLEDGEMENT

The invaluable help extended by Prof. Dr. Soumadip Biswas is appreciated and recognized during this research effort. The Institute of Engineering and Management also offered the resources which enabled this work to be accomplished, and they are sincerely acknowledged for that.

REFERENCES

[1] Feng J., Yadong J., Hao Z., Yan D., Hui L., Shuang M., Yadong W., "Artificial Intelligence in Healthcare: Past, Present and Future", Stroke and Vascular Neurology, 2020, 5 (3), 230–243.
[2] Cai W., "Equipment and Machine Learning in Welding Monitoring: A Short Review", 5th International Conference on Mechatronics and Robotics Engineering (CMRE), Rome, Italy, 2019.
[3] Arjun K., et al., "Web-Based Symptom Checker Using ML", International Journal of Recent Technology and Engineering (IJRTE), 2019, 8, (no issue number specified).
[4] Xia L., et al., "Deep Learning Systems for Disease Diagnosis", Nature Medicine, 2019.
[5] Rajendra C.D., "Machine Learning in Medicine", Circulation, 2015, 132 (20), 1920–1930.
[6] Andre E., Brett K., Roberto A.N., Justin K., Susan M.S., Helen M.B., Sebastian T., "Dermatologist-Level Classification of Skin Cancer with Deep Neural Networks", Nature, 2017, 542 (7639), 115–118.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY