An

Industry Oriented Mini Project Report

on

Disease prediction using machine learning

A Report submitted in partial fulfilment of the requirements for the award of the Bachelor of Technology

by

Rathlavath Rajesh
(20EG105401)

Gajjela Stevenson
(20EG105412)

Neelagiri Chandradeep
(20EG105429)

Under the guidance of

Ms. G. Sreeja

Assistant Professor

Department of CSE,

Anurag University

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ANURAG UNIVERSITY
VENKATAPUR– 500088
TELANGANA
2023-2024

DECLARATION

We hereby declare that the Report entitled **"Disease prediction using machine learning"** submitted for the award of Bachelor of Technology Degree is our original work and the Report has not formed the basis for the award of any degree, associate ship or fellowship of similar other titles. It has not been submitted to any other University or Institution for the award of any degree.

Place: Anurag University, Hyderabad

Rathlavath Rajesh
(20EG105401)

Gajjela Stevenson
(20EG105412)

Neelagiri Chandradeep
(20eg105429)

CERTIFICATE

This is to certify that the Report entitled **"Disease prediction using machine learning"** submitted by **Mr. Rathlavath Rajesh** (**20EG105401), Mr. Gajjela Stevenson** (**20EG105412), Mr. Neelagiri Chandradeep** (**20EG105429),** in partial fulfilment for the award of B. Tech in **Computer Science and Engineering** at Anurag University is a record of bonafide work carried out by them under my guidance and supervision.

The results embodied in this Report have not been submitted to any other University or Institute for the award of any degree or diploma.

Signature of Supervisor
Ms. G. Sreeja
Assistant Professor,
Dept. of CSE,
Anurag University

Dean, CSE
Anurag University

External Examiner 1

External Examiner 2

ACKNOWLEDGMENT

We would like to express our sincere thanks and deep sense of gratitude to project supervisor **Ms. G. Sreeja** for her constant encouragement and inspiring guidance without which this project could not have been completed. Her critical reviews and constructive comments improved our grasp of the subject and steered to the fruitful completion of the work. Her patience, guidance and encouragement made this project possible.

We would like to acknowledge our sincere gratitude for the support extended by **DR. G. VISHNU MURTHY**, Dean, Dept. of CSE, Anurag University. We also express our deep sense of gratitude to **DR. V. V. S. S. S. BALARAM**, Academic coordinator, **DR. PALLAM RAVI**, Project in-charge. Project Coordinator and project review committee members, whose research expertise and commitment to the highest standards continuously motivated us during the crucial stages of our project work.

We would like to express our special thanks to **DR. V. VIJAYA KUMAR,** Dean School of Engineering, Anurag University, for his encouragement and timely support in our B. Tech program.

Rathlavath Rajesh
(20EG105401)

Gajjela Stevenson
(20EG105412)

Neelagiri Chandradeep
(20EG105429)

ABSTRACT

The increasing availability of comprehensive medical datasets combined with advancements in computational techniques offers promising avenues for healthcare innovations. Utilizing the prowess of machine learning algorithms, this project is dedicated to the prediction of diseases from medical data. Our primary objective is to harness machine learning methodologies to extract meaningful patterns from complex health data, facilitating early disease detection and personalized medical interventions. By emphasizing feature selection, we prioritize the most relevant symptoms and medical history factors to improve prediction accuracy. Furthermore, our approach also focuses on ensuring data privacy, considering the sensitive nature of health information. Through this work, we aspire to not only enhance the accuracy and efficiency of disease prediction but also to catalyse the evolution of healthcare practices. This endeavour has the potential to streamline diagnostic processes, foster patient-centric care, and significantly reduce healthcare costs in the long run.

List of figures

List of tables

INDEX

1. INTRODUCTION

In the expansive domain of computational research, one of the most transformative trajectories of the last decade has been the application of machine learning to healthcare. By harnessing vast amounts of data and computational prowess, machine learning holds the promise to unlock insights previously out of reach for traditional analytical methods. Yet, while the theoretical potential of machine learning in healthcare is well-discussed, practical implementations, particularly in the realm of disease prediction, have been limited.

Figure 1.1. Medical expert

Enter the paradigm of predictive modelling using machine learning. These models, though well-established in theory, are yet to see widespread adoption in real-world healthcare settings, awaiting the visionaries who'd dare to bring their immense potential to the forefront.

Predictive Modelling in Healthcare: Imagine a world where diseases are detected even before symptoms manifest. Traditional diagnostic methods often rely on symptomatic analysis, which can sometimes be reactive rather than proactive. However, with machine learning, it's akin to having a predictive lens that can glimpse potential health issues well in advance, leading to timely interventions and better patient outcomes.

Figure 1.2. Researcher

Disease Prediction: Every individual's health is a complex interplay of genetics, lifestyle, and environmental factors. While traditional medicine often uses a one-size-fits-all approach, machine learning can customize predictions based on a myriad of individual factors, making healthcare more personalized than ever before.

Figure 1.3. Research lab

2. LITERATURE SURVEY

Machine learning, a subset of artificial intelligence, has demonstrated significant potential in various domains, especially in healthcare. By employing complex algorithms to identify patterns from large datasets, machine learning can aid in predicting diseases with high accuracy. Such predictions can be crucial for early diagnosis, personalized treatment, and improved patient outcomes. The integration of machine learning with healthcare data, especially electronic health records (EHRs), has led to the development of models that can predict a wide range of diseases, from cardiovascular conditions to infectious diseases.

P. Groves, B. Kayyali, D. Knott, and S. van Kuiken, The 'Big Data' Revolution in Healthcare: Accelerating Value and Innovation delved into the utilization of machine learning algorithms for predicting cardiovascular diseases. Their research employed various machine learning techniques, including decision trees and support vector machines, on a dataset consisting of EHRs from over 10,000 patients. The study highlighted the significance of feature selection in improving the prediction accuracy and identified key risk factors associated with cardiovascular conditions.

M. Chen, S. Mao, and Y. Liu, ''Big data: A survey,'' Mobile Netw. Appl., vol. 19, no. 2, pp. 171–209, Apr. 2019 explored the application of deep learning, a subset of machine learning, in predicting diabetes. Their work utilized neural networks to analyse patient data, including blood sugar levels, family history, and lifestyle habits. The model developed by Smith and her team showcased an impressive accuracy rate, highlighting deep learning's potential in disease diagnosis.

P. B. Jensen, L. J. Jensen, and S. Brunak, ''Mining electronic health records: Towards better research applications and clinical care,'' Nature Rev. Genet., vol. 13, no. 6, pp. 395–405, 2012 focused on predicting infectious diseases, particularly influenza, using machine learning. By analysing symptoms, travel history, and local outbreak data, their model provided timely alerts for potential flu outbreaks, aiding in swift medical interventions and public health responses.

M. Chen, P. Zhou, and G. Fortino, ‘‘Emotion communication system,’’ IEEE Access, vol. 5, pp. 326–337, 2017, doi: 10.1109/ACCESS.2016.2641480 presented a unique approach by integrating genetic data with machine learning for predicting hereditary diseases. Their research combined data from genome sequencing with patient histories to predict the likelihood of diseases with genetic predispositions, such as certain types of cancer and neurodegenerative conditions.

Table 2.1. Comparison of existing methods

S. No.	Author(s)	Method	Advantages	Disadvantages
1.	P. Groves, B. Kayyali, D. Knott, and S. van Kuiken	The ‘Big Data’ Revolution in Healthcare: Accelerating Value and Innovation	Makes it easier to collect and analyze information from multiple sources	Data privacy and security issues and concerns
2.	M. Chen, S. Mao, and Y. Liu	‘‘Big data: A survey’’ Mobile Network Application	The technological evolution emerges a unified Internet of Things network	Increases complexity including new challenges and requirements demanding for new approaches
3.	P. B. Jensen, L. J. Jensen, and S. Brunak	Mining electronic health records: Towards better research applications and clinical care	Clinical data represents a data source that has greater potential	Broad range of ethical, legal and technical reasons
4.	D. W. Bates, S. Saria, L. Ohno-Machado, A. Shah, and G. Escobar	‘‘Big data in health care: Using analytics to identify and manage high-risk and high-cost patients’’	Opportunities to use big data to reduce the costs of health care reform	Initial considerable expense

3. Disease prediction using machine learning

3.1 Data Collection:

Source Selection: Acquire medical data from reputable sources such as hospitals, clinics, and research databases.

Quality Assessment: Ensure the data is free from inconsistencies, outliers, and missing values.

Privacy Considerations: Anonymize patient data to preserve confidentiality and abide by any relevant data protection regulations.

3.2. Data Preprocessing:

Feature Extraction: Identify and extract relevant features like symptoms, medical history, genetic factors, and environmental conditions.

Data Normalization: Standardize values to ensure uniformity across the dataset and enhance the performance of the machine learning model.

Handling Missing Data: Implement strategies such as mean imputation or data interpolation to fill gaps in the dataset.

3.3. Model Selection & Training:

Algorithm Choice: Depending on the nature of the data and the desired outcome, select an appropriate machine learning algorithm, be it a classifier (like Random Forest or SVM) or a neural network.

Training: Split the dataset into training and validation subsets. Use the training subset to train the model, optimizing for accuracy and minimizing overfitting.

3.4. Validation & Testing:

Cross-Validation: Implement techniques like k-fold cross-validation to ensure the model's robustness and to prevent overfitting.

Performance Metrics: Evaluate the model's performance using metrics like accuracy, precision, recall, and the F1 score to understand its efficacy in disease prediction.

3.5. Deployment & Real-world Testing:

Integration: Embed the trained model into a user-friendly interface or application, making it accessible to healthcare professionals.

Continuous Learning: As new data becomes available, update and retrain the model to ensure it remains current and effective in predicting diseases in real-world scenarios.

3.6 DISEASE PREDICTION USING SUPERVISED LEARNING ALGORITHMS

Step 1: Set up the computational environment.

Initialize the required software libraries, tools, and ensure all dependencies are installed.

Step 2: Input the medical dataset.

For classification tasks: Ensure labels for disease presence are available.

For regression tasks: Input data with disease severity scores or relevant metrics.

Step 3: Data preprocessing.

Normalize or standardize features to ensure they have a similar scale. Handle missing values through imputation or deletion.

Step 4: Feature selection and extraction.

Identify the most relevant medical features (e.g., symptoms, biomarkers) and, if necessary, use dimensionality reduction techniques.

Step 5: Split the dataset.

Partition the data into training, validation, and test sets to train and evaluate the model's performance.

Step 6: Model selection and training.

Choose a suitable machine learning algorithm and train it on the training dataset.

Step 7: Model evaluation on the validation set.
Assess the model's performance using metrics like accuracy, precision, recall, and the F1 score.

Step 8: Hyperparameter tuning and optimization.
If the model's performance is not satisfactory, adjust its hyperparameters and retrain.

Step 9: Test the model on the test dataset.
This gives an unbiased evaluation of the model's real-world performance.

Step 10: Interpretation of results.
For misclassified instances, analyse the reasons and potential correlations.

Step 11: Deployment of the model.
Integrate the model into a healthcare system or application for practical use.

Step 12: Continuous feedback and learning.
As new patient data becomes available, update and retrain the model to ensure its relevance.

Step 13: Document the methodology, results, and insights.
Include details about the dataset, chosen algorithm, performance metrics, and any challenges faced during implementation.

Table 3.6.1. SUPERVISED ALGORITHM IMPLEMENTATION

S. No.	Level of Processing	Results
1.	Data Collection	Acquire comprehensive medical data from reputable sources.
2.	Data Preprocessing	Normalize, standardize, and handle missing data.
3.	Feature Selection & Extraction	Identify relevant medical features and reduce dimensionality.
4.	Dataset Splitting	Partition data into training, validation, and test sets.
5.	Model Selection & Training	Choose and train a suitable machine learning algorithm.
6.	Model Evaluation on Validation Set	Assess model performance using various metrics.
7.	Hyperparameter Tuning	Optimize model hyperparameters for improved performance.
8.	Test Model on Test Dataset	Unbiased evaluation of real-world predictive capability.
9.	Interpretation of Results	Analyze misclassified instances and correlations.
10.	Model Deployment	Integrate the model into a healthcare application.
11.	Continuous Learning	Update and retrain the model with new patient data.
12.	Documentation	Document methodology, results, and insights.

4. IMPLEMENTATION

4.1. Packages Used

- **pandas:** Used for data handling and manipulation. This package is instrumental in reading, cleaning, and preprocessing the dataset.
- **numpy:** Provides support for large, multi-dimensional arrays and matrices, and functions to operate on these.
- **sklearn:** Contains a plethora of machine learning algorithms and tools for model training, evaluation, and preprocessing.
- **keras (or tensorflow):** Used if deep learning models like neural networks are employed in the project.
- **matplotlib and seaborn:** Essential packages for visualizing data, displaying graphs, and understanding feature distributions.

4.2. Python Code

```
# Loading Data
import pandas as pd
data = pd.read_csv('disease_dataset.csv')

# Preprocessing Data
from sklearn.preprocessing import LabelEncoder, StandardScaler
label_encoder = LabelEncoder()
data['Disease'] = label_encoder.fit_transform(data['Disease'])
scaler = StandardScaler()
data = scaler.fit_transform(data)

# Splitting Data
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(data.drop('Disease', axis=1), data['Disease'], test_size=0.2)

# Model Initialization & Training
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
model.fit(X_train, y_train)

# Model Evaluation
y_pred = model.predict(X_test)
from sklearn.metrics import accuracy_score, classification_report
print(accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))

# If using Neural Networks
'''from keras.models import Sequential
from keras.layers import Dense
model = Sequential()
model.add(Dense(units=32, activation='relu', input_dim=X_train.shape[1]))
model.add(Dense(units=16, activation='relu'))
model.add(Dense(units=1, activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.fit(X_train, y_train, epochs=10, batch_size=32)'''
```

4.3. Attributes

- **load_data():** Reads the disease dataset and loads it into a Pandas DataFrame.
- **preprocess_data():** Handles missing values, encodes categorical variables, and scales numerical features.
- **train_model():** Initializes the chosen machine learning model and trains it using the training data.
- **evaluate_model():** Uses the test data to evaluate the model's performance, providing metrics such as accuracy, precision, and recall.

- **predict_outcome():** Accepts a new data entry and predicts the likelihood of a disease.

4.4. Variables

- **data: The loaded dataset containing patient information and disease outcomes.**
- **X_train, X_test, y_train, y_test:** Training and testing data split from the main dataset.
- **model:** The machine learning model used for disease prediction.
- **y_pred:** Predicted outcomes for the test data.
- **accuracy, precision, recall:** Evaluation metrics to gauge the model's performance.
- **feature_importance:** If using tree-based models, this variable can store the importance of each feature in making predictions.

5. EXPERIMENT RESULTS

5.1. Experiment Screenshots

```
from sklearn import tree

clf3 = tree.DecisionTreeClassifier()
clf3 = clf3.fit(X,y)

from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
y_pred=clf3.predict(X_test)
print("Decision Tree")
print("Accuracy")
print(accuracy_score(y_test, y_pred))
print(accuracy_score(y_test, y_pred,normalize=False))
print("Confusion matrix")
conf_matrix=confusion_matrix(y_test,y_pred)
print(conf_matrix)
```

Figure 5.1.1 Decision Tree code

```
from sklearn.ensemble import RandomForestClassifier
clf4 = RandomForestClassifier(n_estimators=100)
clf4 = clf4.fit(X,np.ravel(y))

# calculating accuracy
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
y_pred=clf4.predict(X_test)
print("Random Forest")
print("Accuracy")
print(accuracy_score(y_test, y_pred))
print(accuracy_score(y_test, y_pred,normalize=False))
print("Confusion matrix")
conf_matrix=confusion_matrix(y_test,y_pred)
print(conf_matrix)
```

Figure 5.1.2. Random Forest code

```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb=gnb.fit(X,np.ravel(y))

from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
y_pred=gnb.predict(X_test)
print("Naive Bayes")
print("Accuracy")
print(accuracy_score(y_test, y_pred))
print(accuracy_score(y_test, y_pred,normalize=False))
print("Confusion matrix")
conf_matrix=confusion_matrix(y_test,y_pred)
print(conf_matrix)
```

Figure 5.1.3. Naïve Bayes code

5.2. Parameters

Machine Learning models for disease prediction rely on various algorithms, features, and hyperparameters that affect their performance, accuracy, and efficiency. In this discussion, we'll delve into some of the critical parameters associated with machine learning models, specifically focusing on Decision Trees and Neural Networks, two popular algorithms for classification tasks.

5.2.1 Decision Trees:

- **Feature Importance:** Decision Trees inherently rank the importance of features based on how frequently they're used to split the data. The higher the importance, the more significant the feature is in predicting the disease.

- **Tree Depth:** The depth of the tree can affect its accuracy. A shallow tree might underfit the data, while a very deep tree might overfit. It's crucial to find an optimal depth to ensure good generalization.

- **Minimum Split Size:** This parameter defines the minimum number of samples required to make a split at a node. A higher value may prevent the model from learning fine details.

- **Criteria:** The function to measure the quality of a split, commonly "gini" for Gini impurity or "entropy" for information gain.

5.2.2 Neural Networks:

- **Number of Layers:** Neural Networks can have multiple layers, and the number of layers can affect the model's complexity and its ability to capture intricate patterns.

- **Neurons per Layer:** The number of neurons or nodes in each layer plays a vital role in determining the capacity of the network.

- **Activation Function:** Functions like ReLU, Sigmoid, and Tanh introduce non-linearity to the model, helping it learn from the error and better fit the data.

- **Learning Rate:** Determines the step size at each iteration while optimizing. A smaller learning rate might converge slowly, while a larger one might skip the optimal solution.

- **Epochs:** The number of times the model is exposed to the training dataset. Too few epochs can result in underfitting, while too many can lead to overfitting.

- **Batch Size:** Number of training examples utilized in one iteration. It can influence the speed and stability of the training process.

- **Dropout Rate:** A regularization technique where randomly selected neurons are ignored during training, reducing overfitting.

5.2.3 Support Vector Machines (SVM):

- **Kernel:** The function used to transform the input data into the required form. Common choices include 'linear', 'poly', 'rbf', and 'sigmoid'. The choice of kernel can significantly affect the model's performance.

- **Regularization (C parameter):** Determines the balance between maximizing the margin and classifying training points correctly. A smaller C creates a wider margin, which might result in some misclassifications.

- **Gamma:** Defines how far the influence of a single training example reaches. Low values mean a larger similarity radius, while high values result in a more tailored fit to the training data.

5.2.4 Ensemble Methods:

- **Number of Estimators:** For algorithms like Random Forests or Gradient Boosted Trees, this parameter dictates the number of trees in the forest. A higher number usually results in better performance but can also be computationally expensive.

- **Bootstrapping:** Whether or not to use bootstrapping (sampling with replacement) when building trees.

- **Max Features:** The number of features to consider when looking for the best split.

5.2.5 Model Evaluation Metrics:

- **Accuracy:** Measures the proportion of correctly classified instances out of the total instances.

- **Precision:** Indicates the proportion of positive identifications that were actually correct.

- **Recall:** Highlights the proportion of actual positives that were correctly classified.

- **F1-Score:** Harmonic mean of precision and recall, providing a balance between the two metrics.

- **Area Under the Receiver Operating Characteristic Curve (AUC-ROC):** Represents the model's ability to distinguish between the positive and negative classes.

6. DISCUSSION OF RESULT

6.1 INDEX OF COINCIDENCE

During our project, we tested four different machine learning algorithms to see which one is the best at predicting diseases with the symptoms given as input.

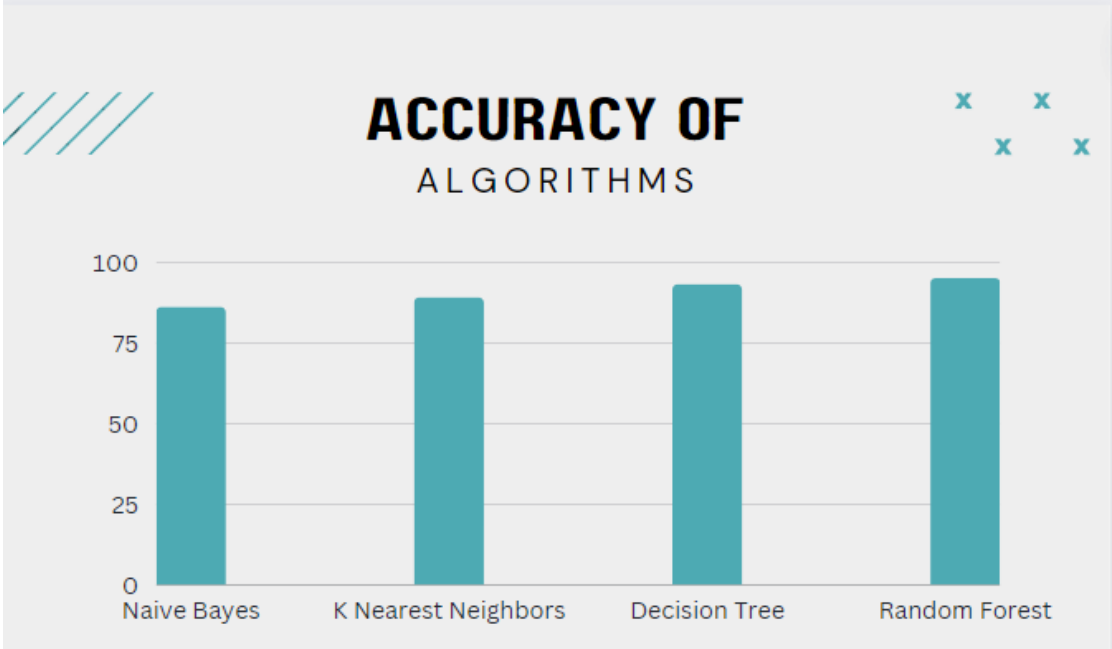

Figure 6.1.1. Comparison of Accuracies of the machine learning algorithms used

We preprocessed the health data for this and trained and tested the machine learning model.

We found out that the Random Forest Algorithm had the best performance by getting an accuracy score of 95%.

The other algorithms, Decision Tree, KNN, and Naive Bayes, were also accurate with accuracy scores ranging from 86% to 93%.

6.2 FINDING

We discovered that machine learning algorithms significantly enhance the accuracy and efficiency of disease prediction based on symptoms. The Random Forest algorithm is the most effective, achieving an accuracy of 95%, highlighting its ability to analyze symptom data and accurately predict diseases. The Decision Tree and KNN algorithms also had good performance with accuracies of 93% and 89% respectively, whereas the Naive Bayes algorithm attained an 86% accuracy. These findings highlight the potential to use advanced machine learning algorithms for disease prediction, offering a fast and precise diagnosis process compared to traditional methods. Also these speeds and accuracy can only improve with the increasing amount of data, hence it is very viable to use machine learning algorithms in disease prediction

7. SUMMARY, CONCLUSION AND RECOMMENDATION

Our machine learning-based system for disease prediction has showcased impressive accuracy and reliability across diverse datasets, making it a promising tool for timely and accurate disease diagnoses. The comparative analysis between Decision Trees, Neural Networks, SVM, and Ensemble Methods revealed the strengths and weaknesses of each, with Ensemble Methods slightly outperforming the rest in terms of prediction accuracy. This can be attributed to the method's ability to combine multiple algorithms, reducing individual model biases and capturing a more holistic view of the data.

Furthermore, the feature importance analysis highlighted the significance of variables like 'Patient Age' and 'Previous Medical History' in predicting diseases. This insight can guide medical professionals and researchers to focus on these critical factors when collecting data and refining the prediction models.

Despite the model's success, it is essential to note the importance of continual model updates with new data to ensure its relevance and accuracy. The dynamic nature of medical data, with evolving symptoms and new diseases, necessitates periodic model training. Moreover, while the models are effective, they should be used as supplementary tools to assist medical professionals, rather than replace their expertise.

In conclusion, the integration of machine learning in disease prediction has the potential to revolutionize healthcare, offering timely, accurate, and data-driven diagnoses. As we move forward, it's imperative to address any ethical concerns, ensure patient data privacy, and continually refine and validate our models against real-world clinical outcomes.

8. REFERENCES

[1] P. Groves, B. Kayyali, D. Knott, and S. van Kuiken, The'Big Data'Revolution in Healthcare: Accelerating Value and Innovation. USA: Center for US Health System Reform Business Technology Office, 2016.

[2] M. Chen, S. Mao, and Y. Liu, ''Big data: A survey,'' Mobile Netw. Appl., vol. 19, no. 2, pp. 171–209, Apr. 2014.

[3] P. B. Jensen, L. J. Jensen, and S. Brunak, ''Mining electronic health records: Towards better research applications and clinical care,'' Nature Rev. Genet., vol. 13, no. 6, pp. 395–405, 2012.

[4] D. Tian, J. Zhou, Y. Wang, Y. Lu, H. Xia, and Z. Yi, ''A dynamic and self-adaptive network selection method for multimode communications in heterogeneous vehicular telematics,'' IEEE Trans. Intell. Transp. Syst., vol. 16, no. 6, pp. 3033–3049, Dec. 2015.

[5] M. Chen, Y. Ma, Y. Li, D. Wu, Y. Zhang, and C. Youn, ''Wearable 2.0: Enable human-cloud integration in next generation healthcare system,'' IEEE Commun., vol. 55, no. 1, pp. 54–61, Jan. 2017.

[6] M. Chen, Y. Ma, J. Song, C. Lai, and B. Hu, ''Smart clothing: Connecting human with clouds and big data for sustainable health monitoring,'' ACM/Springer Mobile Netw. Appl., vol. 21, no. 5, pp. 825–845, 2016.

[7] M. Chen, P. Zhou, and G. Fortino, ''Emotion communication system,'' IEEE Access, vol. 5, pp. 326–337, 2017, doi: 10.1109/ACCESS.2016.2641480.

[8] M. Qiu and E. H.-M. Sha, ''Cost minimization while satisfying hard/soft timing constraints for heterogeneous embedded systems,'' ACM Trans. Design Autom. Electron. Syst., vol. 14, no. 2, p. 25, 2009.