



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** X **Month of publication:** October 2026

DOI: <https://doi.org/10.22214/ijraset.2026.85150>

www.ijraset.com

Call: ☎ 08813907089

E-mail ID: ijraset@gmail.com

AI-Based Adaptive Genome Digital Twin for Personalized Health Prediction

Dr. S. Senthil Kumar¹, Rajeshwarar M², Mohamed Shahil M³

Dept. of Computer Science and Engineering, K.L.N. College of Engineering, Tamil Nadu, India

Abstract: Doctors usually get involved after a patient already feels ill. That is a poor fit for chronic disease, since the damage builds up quietly for years before anyone notices. We built a genome digital twin to deal with this. It is a virtual copy of one person, started from their inherited disease risk and updated with their vital signs, sleep, diet and activity. The inherited part is a polygenic risk score and it never changes. Everything else arrives as new readings and is merged into the twin each time. Random Forest, XGBoost and a Decision Tree baseline take the combined record and give the probability of chronic disease. Results go to a Streamlit dashboard that shows a risk percentage, a biological age, a health score and a wellness index. It also splits the risk into genetic, lifestyle, environmental and clinical shares, and a user can change a habit to watch the forecast move. The paper describes the requirements, a four-layer architecture, how the twin is kept current, the training and evaluation plan, the tests and a twelve-week schedule. We also discuss the limits of the work and the care that genomic data demands. What we think is new is simple: the genetic baseline, the live readings and the what-if simulation sit inside one model, not three separate tools.

Keywords: digital twin, precision medicine, polygenic risk score, multimodal data fusion, ensemble learning, biological age, computational biology

I. INTRODUCTION

A. Background and Motivation

Think about how most people meet the health system. Something hurts, they book an appointment, and a diagnosis follows. Hypertension, type 2 diabetes and coronary disease rarely work like that. They creep up. By the time anything feels wrong, the disease is usually well established and costly to manage. Ginsburg and Phillips [1] argue that precision medicine can bring the decision point forward, because it draws on a person's genome, surroundings and behavior.

Two things make that realistic now. Genotyping has become cheap, so a person's inherited susceptibility can be boiled down to one number. Wearables, meanwhile, log heart rate, blood pressure, blood oxygen and sleep without the wearer doing much at all. Machine learning can then relate one kind of data to another, which Topol [2] discusses in the context of diagnosis. Still, there is no single object in everyday practice that holds all of this, stays current and can be queried.

A digital twin could be that object. The term comes from engineering, where a virtual model follows a physical asset so people can study the asset without disturbing it [3]. Applied to a person, the twin is a living model, not a file in a folder. We can update it, ask it questions, and run *in silico* experiments on it with no risk to the real person. Björnsson et al. [4] argue that this can personalize medicine, since a twin built from a patient's data can project how that patient's disease might progress.

B. Problem Statement

Right now each tool shows only a slice. A genetic report tells you about inherited risk, but it is written once and never updated. A wearable knows how the body is doing today and knows nothing about the genes behind it. A model trained on hospital records has neither view. So the gap is a single model that keeps refreshing, joins the fixed genetic baseline to changing physiology and behavior, and forecasts chronic disease risk. It should also let the person ask what would happen if they changed a habit.

C. Objectives

We set four objectives. We want to shift care from reacting to predicting and preventing. We want to estimate an individual's chronic disease risk by combining static genetic markers with changing vital signs and lifestyle data. We want supervised learning to work on that combined record, so the interaction between genes and daily habits is modeled and not ignored. And we want the results shown visually, so clinicians and patients can follow long-term trajectories and act early.

D. Contributions and Novelty

The contributions of this work are listed below.

- Unified genome-plus-physiology model. Genetic risk and live health data go into one feature vector for each prediction. Nobody has to score them separately and merge the answers by hand.
- Adaptive update cycle. We do not retrain the twin on every reading. A smoothing step folds each new observation into a running state. The estimate follows the person, yet one odd reading does not throw it off (Section V-C).
- Four-way risk decomposition. Each estimate is divided into genetic, lifestyle, environmental and clinical shares. A bare percentage becomes something a person can act on.
- Counterfactual what-if simulation. A user changes a habit and sees how risk and biological age would shift. Prediction is then tied directly to prevention.
- Reproducible open-source prototype. Everything runs on free tools and a synthetic PRS, so the approach can be tried out before real genotype data are in hand.

E. Organization of the Paper

Here is how the paper is laid out. Section II covers the underlying concepts and Section III reviews related work. Section IV compares current systems with ours. Sections V and VI describe the architecture and the requirements, and Section VII gives the methodology. Section VIII shows the implementation and sample outputs, while Section IX presents the test plan. Section X discusses strengths, limitations and ethics. Section XI gives the schedule and Section XII concludes.

II. BACKGROUND CONCEPTS

A. Digital Twins in Healthcare

A digital twin needs three things: a physical entity, a virtual model, and a two-way data link that keeps the two in agreement. In our case the entity is a person, with their habits and environment. The model is a set of learned relationships. The link is the flow of readings from devices, forms and clinical records. What separates a twin from an ordinary predictive model is how long it lives and how it gets used. It is meant to last months or years, to change as the person changes, and to be consulted many times. Corral-Acero et al. [5] describe this idea for cardiology, where a twin might help pick a therapy before anything is tried on the patient.

B. Polygenic Risk Scores

Most common diseases are not caused by a single faulty gene. Instead, hundreds or thousands of small variants each push risk a little up or down. A polygenic risk score (PRS) adds those pushes into one figure. Take g_i as the number of copies of risk allele i that a person carries, and β_i as that allele's effect size from a large association study. The score is the weighted sum in (1). Khera et al. [6] found that such scores can identify people whose risk for conditions like coronary artery disease matches that of rare single-gene mutations.

$$\text{PRS} = \sum_{i=1}^m \beta_i g_i \quad (1)$$

A PRS describes inherited susceptibility. It is not a diagnosis. It also stays the same for life, which makes it a sensible baseline for the twin. We come back to its limits in Section X.

C. Multimodal Health Data

The model uses four kinds of data. Genomic data give the baseline. Physiological data, such as heart rate, blood pressure, blood oxygen and sleep, show the body's present state. Lifestyle data record diet, physical activity and habits. Clinical data hold age, gender, body-mass index and medical history. Hasin et al. [7] note that no single layer explains complex disease, and that combining layers is what exposes mechanisms and risk. The twin follows the same principle, only on a smaller scale.

D. Biological Age

Chronological age is just the years since birth. Biological age tries to say how old the body behaves, and two people born on the same day can differ quite a bit. Horvath [8] built a molecular age estimator from DNA methylation. We do not measure methylation. Our prototype takes a more modest route and derives an indicator from the fused risk profile, then shows it beside the chronological age. A person can see at a glance whether the body looks older or younger than the calendar says.

III. LITERATURE REVIEW

A. Precision Medicine and Genomics

Ginsburg and Phillips [1] followed precision medicine as it grew from a research goal into practical tools, and they stressed that genomic information is central to judging who is susceptible to which disease. They also raised a fair question about value: do personalized methods give enough benefit to justify what they cost? That shaped our design. The twin uses cheap, widely available inputs so it stays affordable to adopt.

B. Artificial Intelligence in Diagnosis

Topol [2] looked at how deep learning is changing diagnosis, particularly when multi-omics data and wearable sensor streams are used together to catch risk early. Rajkomar et al. [9] wrote a wider account of machine learning in medicine. They warned that models need careful evaluation and must be deployed with bias and clinical workflow in mind. Both papers back the use of learned models for risk estimation, and both caution against treating model output as certain. That is why our dashboard shows probabilities and ranges, not verdicts.

C. Digital Twins for Physiology and Disease

Björnsson et al. [4] proposed a digital-twin framework for human physiology. They showed that simulation models built from real patient data can forecast how a disease will run its course. Corral-Acero et al. [5] outlined a program for cardiac twins that brings together imaging, electrophysiology and machine learning. Both works show that patient-specific virtual models are feasible. Both also stay at the level of an organ or a single disease. Neither treats the inherited genetic baseline and everyday lifestyle as inputs that evolve together in a general chronic-risk forecast.

D. Polygenic Risk and Machine Learning

Torkamani et al. [10] reviewed how useful polygenic risk scores are in the clinic. They found that, combined with machine learning, the scores predict chronic disease risk better than family history does. Khera et al. [6] supported this with genome-wide scores for several common diseases. Martin et al. [11] then raised a serious concern. Most scores were developed on people of European ancestry and work less well for other groups, so careless clinical use could widen health gaps. In this line of work the score is normally a stand-alone predictor, and it is rarely revised against current physiology.

E. Research Gap

Three gaps stand out in the literature. Genetic and lifestyle data are usually analyzed apart from each other. Twin frameworks tend to be built around one organ or disease, and they are rarely personalized down to the genome. And few systems update their estimate as the person changes, or let the person test a change in behavior. Our work sits right where these three meet: a continuously updated twin anchored in the genome, with an interactive counterfactual interface.

IV. EXISTING SYSTEMS AND THEIR LIMITATIONS

Today's approaches have five weaknesses. Clinical care is reactive. Diagnosis and treatment come after symptoms, not before. Consumer genetic-testing services hand over a one-time report that never changes, although the customer's lifestyle and physiology do. Fitness trackers capture vitals in real time but never connect them to the wearer's genetic background.

Models in the literature are often trained on one type of data, either genomic or clinical. That limits them for conditions with both inherited and environmental causes. Last, no unified twin exists that keeps evolving and links genetic predisposition to everyday health measurements for long-range forecasting. Table I sets these systems next to the proposed one.

TABLE I. COMPARISON OF EXISTING APPROACHES WITH THE PROPOSED GENOME DIGITAL TWIN

Aspect	Routine clinical care	One-off genetic report	Wearable app	Proposed twin
Timing	After symptoms	Once	Continuous	Continuous
Uses genome	Rarely	Yes	No	Yes
Uses live vitals	Visits only	No	Yes	Yes
Updates over time	Per visit	No	Trend only	Yes, adaptive
Forecasts risk	Limited	Static	Rarely	Yes
What-if testing	No	No	No	Yes

V. PROPOSED SYSTEM

We propose an AI-based adaptive genome digital twin. It puts static genetic markers and dynamic vital-sign and lifestyle data into a single model. Ensemble learners estimate chronic disease risk from it, and the estimate is revised whenever new readings come in. The next subsections describe the architecture, the flow of data, the update rule that makes the twin adaptive, and the modules behind it.

A. System Architecture

The design has four stacked layers, shown in Fig. 1. The data layer gathers genomic markers as a polygenic risk score, wearable and vital readings (heart rate, blood pressure, oxygen saturation and sleep), lifestyle inputs and clinical history. The processing layer cleans that data and fills in missing values. It also scales the features, uses exploratory analysis to pick the informative ones, and fuses all the modalities into one feature set.

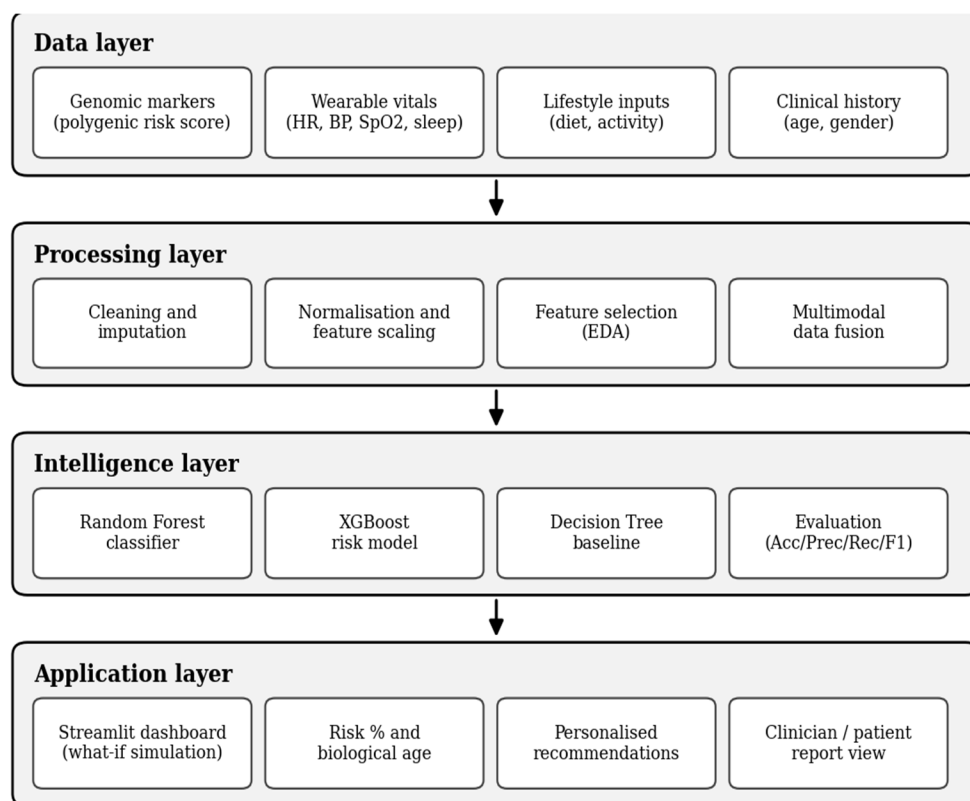


Fig. 1. Layered architecture of the proposed genome digital twin, from data acquisition to the application layer.

The intelligence layer trains and evaluates the Random Forest classifier, the XGBoost risk model and the Decision Tree baseline. The application layer delivers the results. It covers the dashboard, the risk and biological-age indicators, personalized recommendations and a reporting view for clinicians and patients. The layers are kept separate on purpose, so one of them, for example the model or the interface, can be swapped out without touching the rest.

B. Data Flow

Fig. 2 shows the user journey and how data moves through the system. A user registers, signs in, and supplies a genome profile plus dynamic details such as age, gender, BMI, blood pressure, lifestyle habits and medical history. The data are cleaned and normalized and then passed to the trained model. Out come a disease-risk probability and a biological-age estimate, which appear on a personalized dashboard with health recommendations.

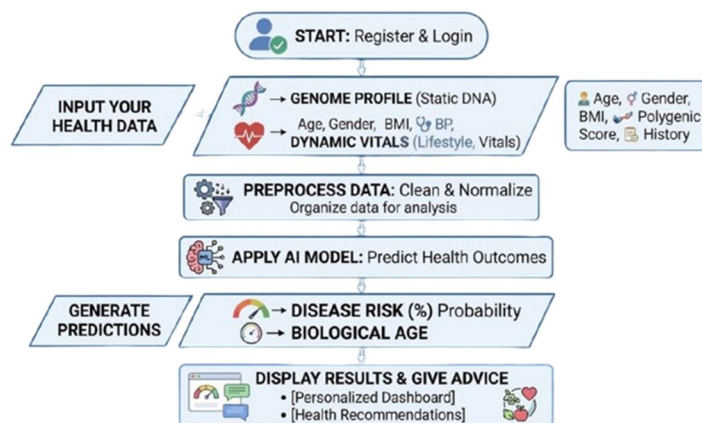


Fig. 2.Data flow of the system, from registration and health-data entry to prediction and advice.

C. Adaptive Twin Update

The cycle in Fig. 3 is what makes the twin adaptive. Every new reading is validated, folded into a running dynamic state, fused with the fixed genomic baseline and scored again. A single reading can mislead, for instance one unusually high blood-pressure value, so we do not simply overwrite the state with the latest number. We use an exponentially weighted average, given in (2). Here x_t is the newest reading of a feature, s_t is its smoothed value, and α , a number between 0 and 1, sets how fast the twin forgets the past.

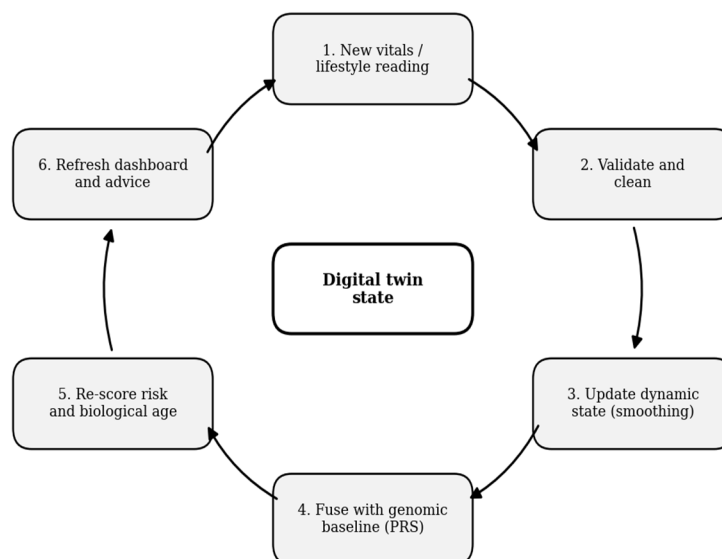


Fig. 3.Adaptive update cycle that keeps the twin synchronized with new observations.

$$s_t = \alpha x_t + (1 - \alpha) s_{t-1} \quad (2)$$

With a large α the twin reacts quickly, which suits signals that move fast, such as heart rate. With a small α it stays steady, which suits slow signals such as weight. The model receives the baseline joined to the smoothed dynamic features, as in (3), where c holds the static clinical attributes. The classifier f then returns a risk probability, as in (4).

$$z_t = [\text{PRS}, s_t, c] \quad (3)$$

$$r_t = f(z_t), \quad 0 \leq r_t \leq 1 \quad (4)$$

The baseline term never changes, but the smoothed term does. So whenever the estimated risk shifts, we can put it down to the person's recent behavior and not to the genome. This is the property behind the four-way risk breakdown in Section VII-H.

D. Module Description

The prototype is divided into six modules.

- Registration and profile. Creates the twin and stores age, gender, BMI, history, and the genomic baseline.
- Data acquisition. Accepts vitals and lifestyle entries and timestamps them.
- Preprocessing. Cleans records, imputes missing values, encodes categories, and scales numeric features.
- Prediction engine. Applies the trained model to the fused vector and returns risk, biological age, health score, and wellness index.
- Simulation engine. Copies the current state, applies the lifestyle changes the user proposes, and re-scores the copy. The original twin is left untouched.
- Presentation and advice. Renders the dashboard, genome view, simulation panel, and recommendations.

VI. SYSTEM REQUIREMENTS AND DESIGN

A. Functional Requirements

Table II lists what the system must do. Each requirement has a number, so the test cases in Section IX can point back to it.

TABLE II. FUNCTIONAL REQUIREMENTS

ID	Requirement
FR1	The system shall let a user register, sign in, and create a personal twin.
FR2	The system shall accept genomic baseline data (polygenic risk score) and clinical attributes.
FR3	The system shall accept vitals and lifestyle inputs and update the twin state.
FR4	The system shall predict individual disease risk as a percentage.
FR5	The system shall estimate and display biological age beside chronological age.
FR6	The system shall divide risk into genetic, lifestyle, environmental, and clinical parts.
FR7	The system shall let the user simulate lifestyle changes and view the projected effect.
FR8	The system shall generate personalized, risk-based recommendations.

B. Non-Functional Requirements

The system must also meet the quality attributes in Table III. We treat privacy as a first-order requirement. Genomic information is very sensitive and, once disclosed, it cannot be changed.

TABLE III. NON-FUNCTIONAL REQUIREMENTS

Attribute	Requirement
Usability	A non-specialist can read the dashboard and run a simulation without training.
Responsiveness	Prediction and simulation results appear within a few seconds of input.
Privacy	Personal and genomic data stay local to the user session and are not shared without consent.
Explainability	Each prediction is accompanied by the factors that drove it.
Maintainability	Layers and modules are decoupled so a model can be swapped without interface changes.
Portability	The system runs on a standard laptop using open-source software only.

C. Use Cases and Modeling

Two kinds of people use the system. The **patient** registers, enters data, views the dashboard, runs what-if simulations and reads the recommendations. The **clinician** opens the reporting view for a patient, looks at the genome panel, and uses the projected trajectory to discuss preventive steps. The main sequence of interaction follows Fig. 2: data entry, preprocessing, prediction and display. The project documentation contains the formal software requirement specification, the data flow diagrams and the UML diagrams (use case, class and sequence) that expand on this summary.

VII. METHODOLOGY

A. Dataset and Genomic Simulation

Health records like the ones we use have physiological and lifestyle attributes, but almost never a matching genome. We did not want to wait for paired genotype and phenotype data, so we injected a synthetic polygenic risk score into every record. It stands in for the genomic baseline, and the experimenter decides how strongly it relates to the outcome. This is a deliberate simplification. It lets us build and test the whole pipeline end to end. Section X explains what we can and cannot claim because of it. Table IV groups the attributes.

TABLE IV. ATTRIBUTE GROUPS USED TO BUILD THE TWIN

Group	Example attributes	Nature
Genomic	Polygenic risk score (synthetic)	Static
Physiological	Heart rate, blood pressure, SpO ₂ , sleep	Dynamic
Lifestyle	Diet, physical activity, habits	Dynamic
Clinical	Age, gender, BMI, history	Mostly static
Target	Chronic disease risk class	Label

B. Preprocessing

Preprocessing drops duplicate rows, checks every column for impossible values, fills missing entries and turns categorical fields into numbers. Continuous features are rescaled to similar ranges, so no single unit dominates in distance-based methods or regularized optimization. Missing values are dealt with column by column. A numeric gap gets the median, which is robust to outliers. A categorical gap gets the most frequent category.

C. Exploratory Analysis and Feature Selection

Exploratory analysis does two jobs. It reveals data-quality problems, such as skewed distributions or class imbalance. It also shows how the static and dynamic variables relate to each other and to the outcome. We use correlation matrices, distribution plots and grouped comparisons here. Features that add little or repeat others are dropped, and the importance scores from the tree ensembles give a second opinion on which variables matter.

D. Data Fusion

We chose early fusion. The genomic score, the smoothed dynamic features and the clinical attributes are joined into the single vector of (3) before any learning happens. Tree ensembles can find interactions between the baseline and the dynamic features on their own, which is why early fusion suits them. A high genetic score together with poor sleep, for example, may carry more risk than either alone.

E. Learning Models

The data are split into training and testing subsets. The split is stratified, so class proportions match in both. We implement three supervised learners.

- Decision Tree. A tree divides the feature space with a series of threshold tests [12]. Each split is picked to reduce impurity, measured for example by the Gini index in (5), where p_k are the class shares at the node. Trees are easy to read, so this one is our interpretable baseline.

$$G = 1 - \sum_k p_k^2 \quad (5)$$

- Random Forest. A set of trees, each trained on a bootstrap sample and limited to a random subset of features at every split [13]. The trees vote on the prediction. Voting lowers variance and copes well with mixed, noisy inputs.
- XGBoost. A gradient-boosted tree ensemble [14], [15]. It adds trees one at a time, and each new tree is fitted to the errors left by the earlier ones. Its objective, (6), balances a loss term l against a penalty Ω on the complexity of each tree, which keeps overfitting in check.

$$L = \sum_j l(y_j, \hat{y}_j) + \sum_k \Omega(f_k) \quad (6)$$

All models are built with scikit-learn [16] and the XGBoost library. The three learners have increasing capacity. That lets us test whether the added complexity of boosting really pays off against a single tree.

F. Hyperparameter Tuning

Hyperparameters of the strongest candidates are tuned by cross-validation, using the training set only. The test set stays untouched until the final comparison. Table V lists the parameters we search. We want a model that is accurate without being over-fitted, and we stop tuning once validation performance levels off.

TABLE V. HYPERPARAMETERS EXPLORED DURING TUNING (SEARCH SPACE, NOT RESULTS)

Model	Parameter	Role
Decision Tree	Max depth, min samples per leaf	Limits tree size
Random Forest	Number of trees, max depth, features per split	Variance and diversity
XGBoost	Learning rate, number of rounds, max depth, subsample	Boosting speed and complexity
XGBoost	L1 / L2 regularization	Overfitting control

G. Evaluation Metrics

Models are compared on accuracy, precision, recall and F1-score. These come from the counts of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN):

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad (7)$$

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad (8)$$

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \quad (9)$$

$$F_1 = 2 \cdot \text{Precision} \cdot \text{Recall} / (\text{Precision} + \text{Recall}) \quad (10)$$

In screening, missing a high-risk person costs more than an extra follow-up. For that reason we watch recall closely, and the F1-score sums up the balance between the two kinds of error. If the classes are unequal, accuracy alone can make a model look better than it is, so we always report the other three metrics with it. Feature attributions in the style of Lundberg and Lee [17] could also show which inputs pushed an individual prediction up or down.

H. Risk Decomposition, Biological Age, and Simulation

Besides the headline risk, the dashboard attributes the estimate to four sources: genetic, lifestyle, environmental and clinical. The fused vector is divided into feature groups, so the contribution of each group can be given as a share of the total. The biological-age indicator comes from the same fused risk profile and is shown against chronological age. The what-if engine works on a copy of the twin. It overwrites chosen lifestyle values with the ones the user proposes, rebuilds the fused vector and scores it again. The gap between the original and the simulated result is the projected effect of the change. Because the genomic baseline is fixed, only the behavioral part of the estimate moves.

VIII. IMPLEMENTATION AND SAMPLE OUTPUTS

A. Technology Stack

The prototype uses open-source tools only, grouped by role in Table VI. Python was our choice for its mature data-science ecosystem. Streamlit turns a Python script into an interactive web interface with very little front-end code. Flask is kept as an optional way to serve the model as a REST-style service.

TABLE VI. TECHNOLOGIES USED IN THE IMPLEMENTATION

Category	Tools	Purpose
Data and ML	Python 3, Pandas, NumPy, Scikit-learn	Data handling and model building
Algorithms	Random Forest, XGBoost, Decision Tree	Disease risk prediction
Analysis	Matplotlib, EDA, feature selection	Insight and feature ranking
Genomic simulation	Synthetic PRS injection	Genomic baseline for each record
Application	Streamlit, Flask (optional API)	Interactive what-if interface
Evaluation and docs	Accuracy, precision, recall, F1; SRS, DFD, UML	Validation and documentation

B. Dashboard and Outputs

Figs. 4–7 are screenshots of the working prototype. The main dashboard (Fig. 4) sums up a user's twin with four indicators: disease risk, biological age against chronological age, an overall health score and a wellness index. In the sample shown, the twin reports an 18% disease risk, a biological age of 16 years for a chronological age of 20, a health score of 87% and a wellness index of 83%. The risk breakdown gives genetic (29%), lifestyle (13%), environmental (16%) and clinical (12%) components, and a trend panel projects life expectancy over the coming decades. These numbers come from a demonstration profile. They illustrate the interface and are not clinical findings.

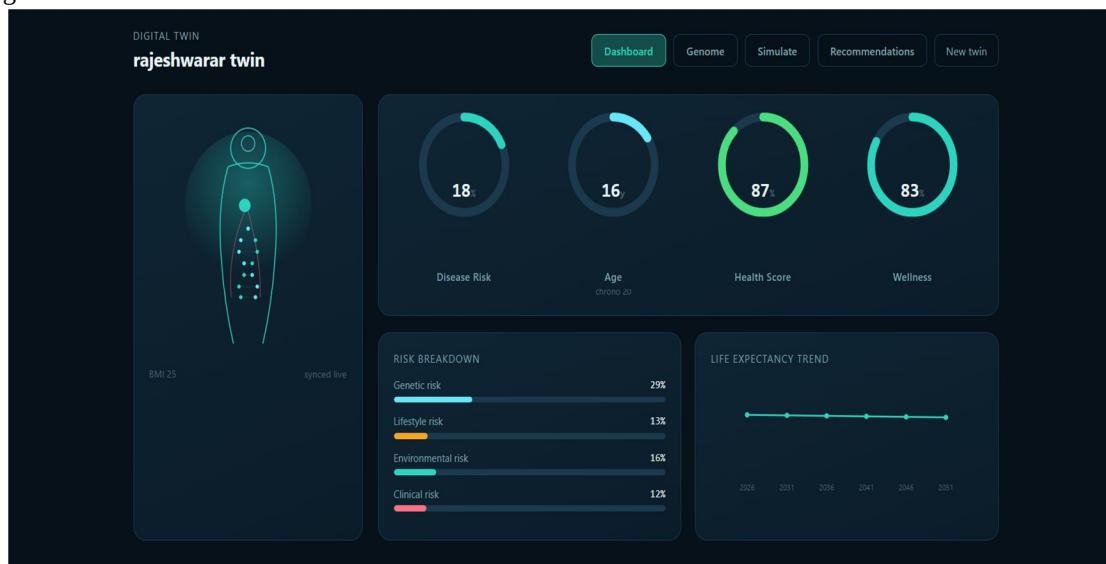


Fig. 4. Digital twin dashboard showing disease risk, biological age, health score, and wellness index computed from the fused genome and vitals profile.

The genome view (Fig. 5) lists the polygenic risk scores and the detailed biological markers behind each individual prediction. Showing these inputs keeps the model's reasoning open to inspection. A clinician can then judge whether the explanation is plausible.

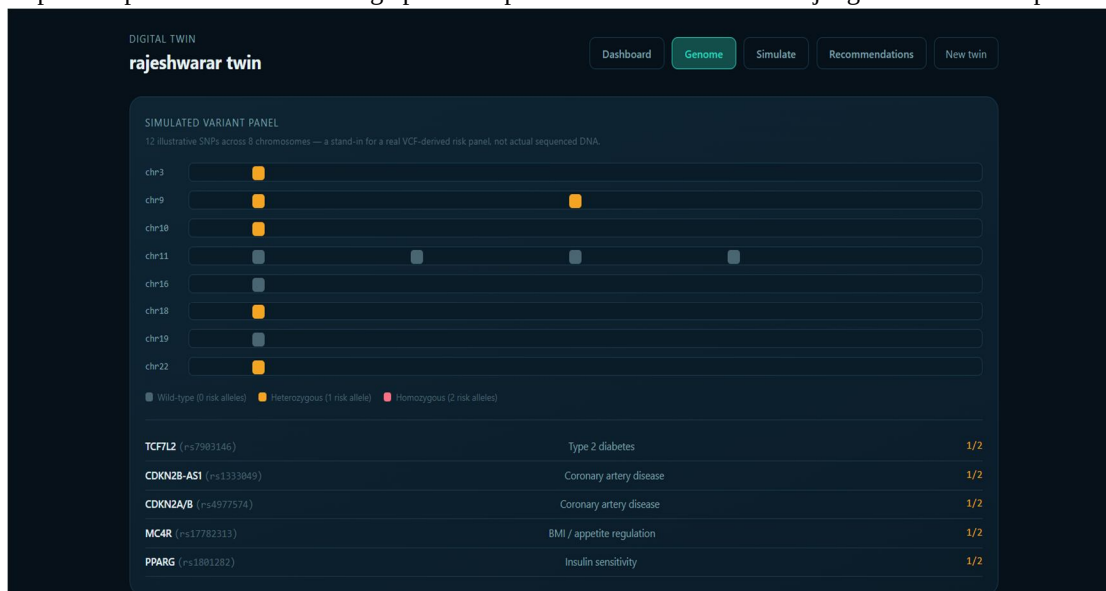


Fig. 5. Genome analysis view listing polygenic risk scores and the markers behind the personalized prediction.

In the simulation panel (Fig. 6) the user changes lifestyle variables and sees the projected effect on long-term health. An abstract percentage becomes a concrete comparison between possible futures. We consider this the feature most likely to change behavior.

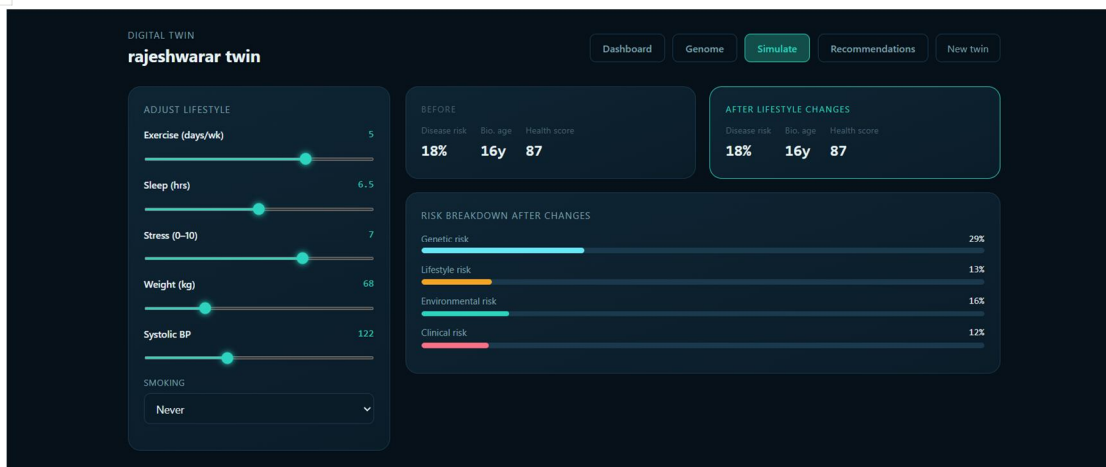


Fig. 6. What-if simulation panel in which lifestyle variables are adjusted to preview their effect on the health trajectory.

Last, the recommendation view (Fig. 7) turns risk levels into practical advice, for example preventive screening or a doctor consultation. The advice follows the risk components. If the lifestyle share dominates, the user is pointed toward habits. If the clinical share dominates, the user is advised to seek medical review.

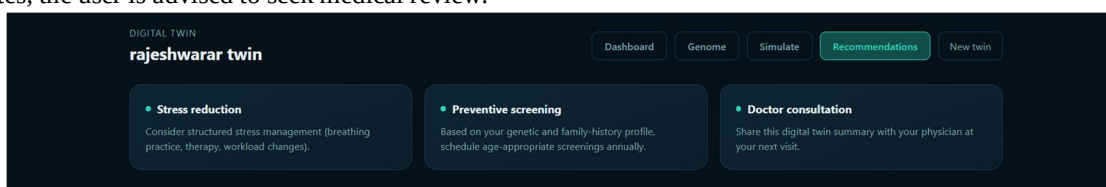


Fig. 7. Personalized, risk-driven recommendations generated by the system.

IX. TESTING AND VALIDATION PLAN

A. Model Validation

Model quality is judged on data the model has not seen. We split the dataset into training and test portions and tune with cross-validation inside the training portion. The final comparison of the three learners uses the held-out test portion once. For each model we will report accuracy, precision, recall and F1-score, together with a confusion matrix, so we can study the type of error and not only its rate. The genomic score is synthetic, so we also plan an ablation. Models will be trained with and without the score to measure how much the genomic baseline adds.

B. System Testing

System testing checks the whole application against the requirements in Tables II and III. We use varied patient records, including normal, borderline and high-risk profiles, and also deliberately incomplete or extreme entries. Table VII lists representative test cases.

TABLE VII. REPRESENTATIVE SYSTEM TEST CASES

ID	Scenario	Expected behavior	Req.
T1	Register and sign in with valid details	Twin is created and dashboard opens	FR1
T2	Enter complete profile with typical values	Risk, biological age, and scores are shown	FR2, FR4, FR5
T3	Enter record with a missing vital	Value is imputed; no crash; prediction shown	FR3
T4	Enter implausible value (e.g., negative heart rate)	Input rejected with clear message	FR3
T5	Increase activity in simulation panel	Projected risk changes; original twin unchanged	FR7
T6	High-risk profile	Recommendations advise clinical review	FR8
T7	Open risk breakdown	Four component shares are displayed	FR6

C. Usability Review

The interface is meant for non-specialists, so we plan a short usability review. A small group of volunteers will do typical tasks, such as reading their risk and running a simulation, and tell us where the wording or layout confused them. What we learn goes back into the interface during the last weeks of the schedule.

X. DISCUSSION

A. Strengths

The main strength of the design is integration. The genetic baseline, current physiology and behavioral simulation sit in one model that is refreshed as data arrive. The person does not have to piece together several disconnected reports. The layered structure keeps the system maintainable and the open-source tools keep the cost low. Since risk is shown as components and what-if testing is available, the output is more than a number. It points at something the person can change.

B. Limitations

We want to be plain about the limitations. First, the genomic baseline is synthetic. The prototype shows that the fusion pipeline works. It cannot show that a real polygenic score improves prediction, because that needs paired genotype and outcome data. Second, published scores are not equally accurate across ancestries [11]. Any real deployment must check that a score is valid for the population in question. Third, the biological-age figure is a derived indicator. It is not a molecular measurement of the kind Horvath [8] describes, so it should be read as a rough summary. Fourth, the what-if projections are model-based associations. They do not prove that a lifestyle change will cause a given outcome.

C. Ethical and Privacy Considerations

Genomic data say something about biological relatives as well as the individual, and they cannot be changed if they leak. A production system would need explicit informed consent, secure storage, minimal retention and openness about how predictions are made. Fairness matters too. A model that works well for one group and badly for another can deepen existing inequalities. Risk also has to be communicated with care, because a high probability shown without context can cause needless anxiety. For these reasons we designed the dashboard as decision support for a person and a clinician together. It is not an autonomous diagnostic tool.

XI. PROJECT PLAN

The work is planned over twelve weeks, summarized in Table VIII and drawn as a chart in Fig. 8. The early weeks fix the scope and the data. The middle weeks build and tune the models. The final weeks go to the interface, testing and documentation.

TABLE VIII. TWELVE-WEEK PROJECT SCHEDULE

Weeks	Activities	Expected outcome
1–2	Problem definition, scope, and proposal; literature survey on digital twins and multi-omics; choice of algorithms	Proposal and literature review complete
3–4	Data collection, synthetic PRS injection, and cleaning; exploratory analysis and feature selection	Clean multimodal dataset; key features chosen
5–6	Train/test split; Random Forest, XGBoost, and Decision Tree models; first Streamlit what-if dashboard	Initial models and dashboard prototype
7–8	Full training, metric comparison, and hyperparameter tuning to limit overfitting	Best optimized model ready for deployment
9–10	User interface with real-time inputs; system testing on varied patient records, validation, and bug fixing	Stable, tested application
11–12	SRS, DFD, UML, results, and user manual; final testing, presentation, and report submission	Documentation done; project complete

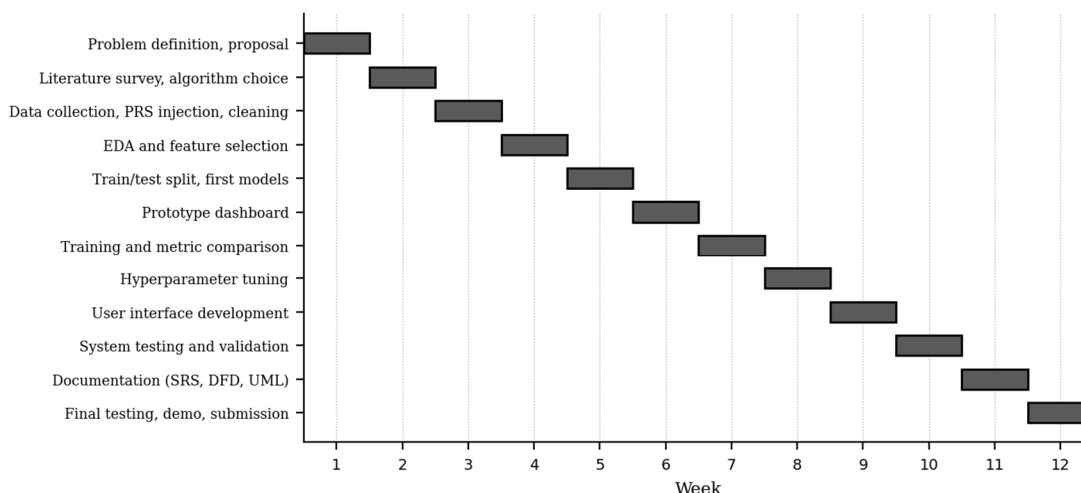


Fig. 8. Gantt-style view of the twelve-week schedule.

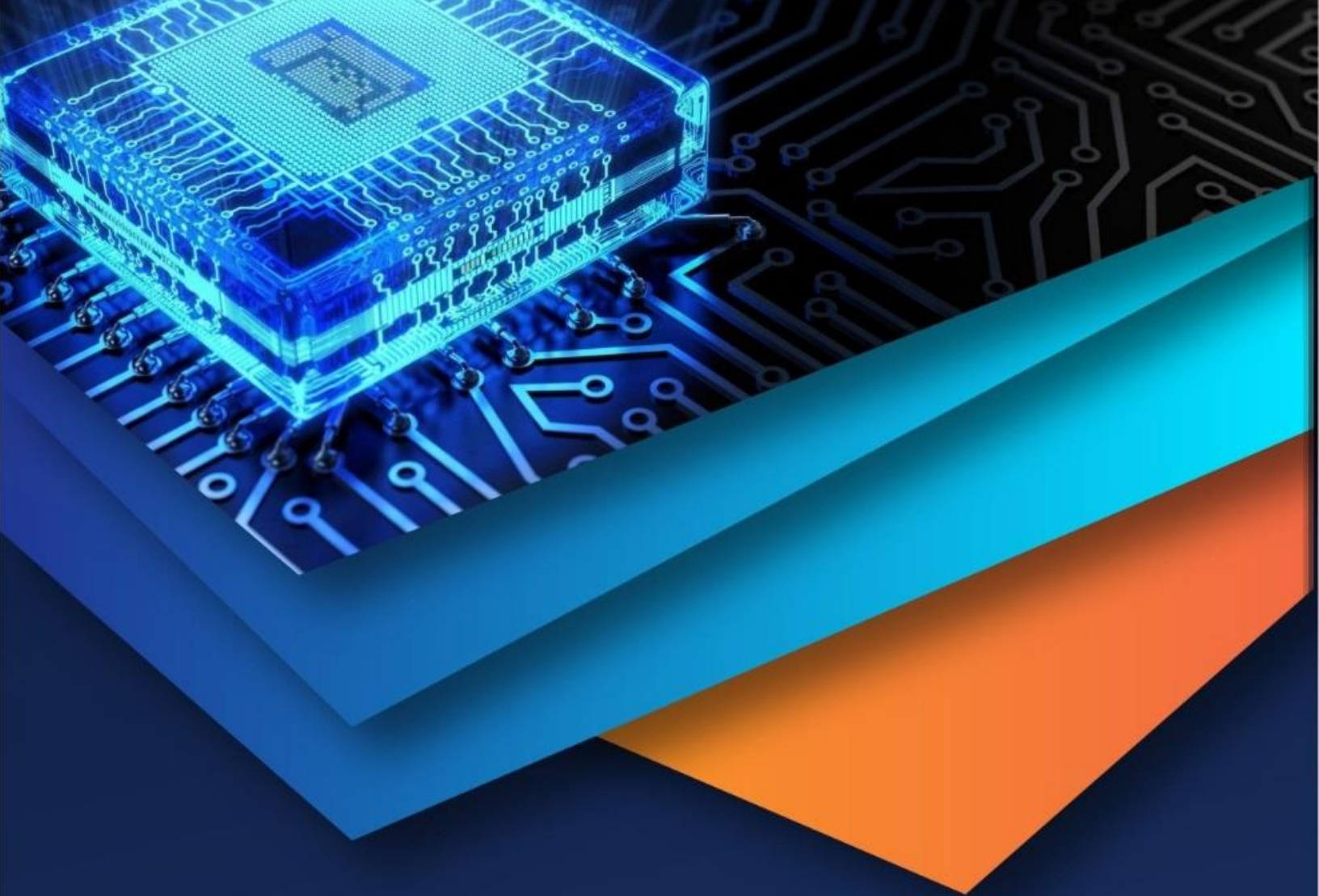
XII. CONCLUSION AND FUTURE WORK

This paper presented an AI-based adaptive genome digital twin. It joins inherited risk with live physiological and lifestyle data to forecast chronic disease for one person. Four ideas support the design: a fixed genomic baseline, a smoothed dynamic state that follows the person, ensemble learners that relate the two, and a what-if interface that links prediction to action. With them, health management can move from reaction toward prevention, and patients and clinicians share one picture of likely futures.

Several next steps are clear. The most important is to replace the synthetic score with real genotype data and check the models against clinical outcomes across different ancestries. Live wearable feeds could replace manual entry. The model could cover more disease categories and use time-series learners that exploit the full history of readings. Feature-attribution methods would make each prediction easier to explain, and federated or on-device learning could keep sensitive data in the user's hands. Prospective studies with clinicians would then show whether seeing a personal twin changes behavior and outcomes.

REFERENCES

- [1] G. S. Ginsburg and K. A. Phillips, "Precision medicine: From science to value," *Health Aff.*, vol. 37, no. 5, pp. 694–701, 2018.
- [2] E. J. Topol, "High-performance medicine: The convergence of human and artificial intelligence," *Nat. Med.*, vol. 25, no. 1, pp. 44–56, 2019.
- [3] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary Perspectives on Complex Systems*, F.-J. Kahlen, S. Flumerfelt, and A. Alves, Eds. Cham, Switzerland: Springer, 2017, pp. 85–113.
- [4] B. Björnsson et al., "Digital twins to personalize medicine," *Genome Med.*, vol. 12, Art. no. 4, 2020.
- [5] J. Corral-Acero et al., "The 'digital twin' to enable the vision of precision cardiology," *Eur. Heart J.*, vol. 41, no. 48, pp. 4556–4564, 2020.
- [6] A. V. Khera et al., "Genome-wide polygenic scores for common diseases identify individuals with risk equivalent to monogenic mutations," *Nat. Genet.*, vol. 50, no. 9, pp. 1219–1224, 2018.
- [7] Y. Hasin, M. Seldin, and A. Lusis, "Multi-omics approaches to disease," *Genome Biol.*, vol. 18, Art. no. 83, 2017.
- [8] S. Horvath, "DNA methylation age of human tissues and cell types," *Genome Biol.*, vol. 14, Art. no. R115, 2013.
- [9] A. Rajkomar, J. Dean, and I. Kohane, "Machine learning in medicine," *N. Engl. J. Med.*, vol. 380, no. 14, pp. 1347–1358, 2019.
- [10] A. Torkamani, N. E. Wineinger, and E. J. Topol, "The personal and clinical utility of polygenic risk scores," *Nat. Rev. Genet.*, vol. 19, no. 9, pp. 581–590, 2018.
- [11] A. R. Martin et al., "Clinical use of current polygenic risk scores may exacerbate health disparities," *Nat. Genet.*, vol. 51, no. 4, pp. 584–591, 2019.
- [12] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Belmont, CA, USA: Wadsworth, 1984.
- [13] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [14] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 2001.
- [15] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, 2016, pp. 785–794.
- [16] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
- [17] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017, pp. 4765–4774.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)