Biomedical Signal Processing and Control 113 (2026) 108702

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc

StaBloCare: Blockchain-secured diabetes monitoring system using stacked deep learning

Monu Bhagat [a,b,*], Ujjwal Maulik [c,d]

[a] Department of Computer Science and Engineering, Birla Institute of Technology, Mesra, Ranchi, India
[b] Department of Artificial Intelligence & AI4ICPS I Hub Foundation, IIT Kharagpur, India
[c] Department of Computer Science & Engineering, Jadavpur University, India
[d] Visiting Professor, Department of Electrical Engineering, Stanford University, USA

ARTICLE INFO

Keywords:
Blockchain
Diabetes
PIDD
XGBoost
RF
SVM
DT
Stacked deep learning

ABSTRACT

Diabetes is a prevalent chronic condition that significantly impacts various aspects of an individual's health. Leveraging machine learning techniques for diabetes prediction offers a promising alternative to traditional diagnostic methods by enhancing both efficiency and accuracy. In this study, we utilize the PIDD to develop more accurate and secure machine learning-based classification models for early diabetes detection. In order to determine if a patient has diabetes, PIDD uses their input data, which includes their blood sugar, blood pressure, skin thickness, insulin level, and BMI. From the dataset, various classification methods are taught, such as DT, KNN, SVM, KSVM, XGBoost, RF Classifier, and a deep learning-based stacking classifier. A number of measures are used to assess the efficiency of each algorithm, including recall, accuracy, F1, precision, and classification report. Stacking these ML classifiers with deep learning ones yields the best results, as shown by the 0.905 accuracy and 0.9211 AuC scores. The suggested model has the ability to include into clinical workflows for early-stage screening systems by predicting the existence of diabetes using cross-sectional data. Using blockchain technology, we have created an impenetrable system that safeguards patient data every step of the way through the prediction process, eliminating the risks associated with traditional centralized systems such as data breaches, illegal access, and manipulation.

1. Introduction

Healthcare is one industry where ML approaches have shown considerable promise in recent years. Effective illness management and better patient outcomes depend heavily on early disease identification. In this study, we use machine learning methods to develop a prediction model for early diabetes identification. However, the investigation of new procedures to protect sensitive medical information is necessary due to the growing concern for data privacy and security. We suggest integrating blockchain technology to improve the security and integrity of the predictive model to allay this worry [1]. Millions of individuals throughout the world suffer from the chronic condition of diabetes. Medical data analysis and prediction using machine learning algorithms has proven to be useful. In our study, we used a variety of ML methods to train and test the predictive model, including Gaussian NB, DT Classifier, KNN, SVM, XGBoost and RF. Utilizing parameters such as accuracy, F1 score, precision, recall, and classification report, the effectiveness of each method was evaluated. Even if the ML model showed promising results, the security of private patient information is still a serious problem. The hazards of data breaches, unauthorized access, and tampering are present with traditional centralized systems. We suggest incorporating blockchain technology into the diabetes early detection system to address these issues. Blockchain provides a decentralized, unchangeable ledger that ensures data integrity, transparency, and trust. We want to develop a safe and impenetrable system that protects patient data throughout the prediction process by utilizing blockchain. By distributing data and decision-making over a network of nodes, blockchain technology enables the removal of the requirement for a centralized authority. A high level of security is provided by the transparent and auditable recording of every transaction and data exchange. A medical disorder known as diabetes occurs when the body's insulin levels are uncontrolled. Insufficient insulin production by the pancreas or an issue with insulin resistance in the cells constitutes diabetes mellitus type 2. Insulin controls blood sugar levels [2]. Among persons who were 18 years old and above, 8.5% had diabetes in 2014. In 2014, those under the age of 70 accounted for 48% of all fatalities,

* Corresponding author at: Department of Artificial Intelligence & AI4ICPS I Hub Foundation, IIT Kharagpur, India.
E-mail addresses: monu.t24@kgpian.iitkgp.ac.in (M. Bhagat), ujjwal.maulik@jadavpuruniversity.in (U. Maulik).

https://doi.org/10.1016/j.bspc.2025.108702
Received 14 March 2025; Received in revised form 31 July 2025; Accepted 18 September 2025
Available online 30 October 2025

Table 1
Centralized vs Blockchain Logging.

Feature	HIPAA-Compliant Centralized	Blockchain-Based Logging
Write speed	High	Moderate
Tamper resistance	Relies on audit trails	Inherent via immutability
Audit trail	External system needed	Built-in via hash chain
Regulatory compliance	HIPAA, but mutable	GDPR/HIPAA-compatible design
Cost per transaction	Low (shared infrastructure)	Moderate (gas fees)

with diabetes being the leading cause of 1.5 million deaths [3]. According to WHO, there are around 102 million adults in India who are either diabetic or prediabetic (at higher risk of contracting the disease). Major issue that India is facing is that more than half of the people are unaware of their diagnosis. This delay in diagnosis leads to major health complications. Adults with diabetic condition have an increases risk of heart related ailments and strokes. The effect of long-term diabetes is threatening for the human body; it effects major organs of the body. Hence, people at high risk can take preventative measures to delay the disease's advancement and enhance their quality of life when diagnosed early [2,4].

1.1. HIPAA and blockchain

HIPAA [5] provides regulatory standards for safeguarding sensitive patient data in centralized electronic health systems. While HIPAA-compliant centralized architectures support high throughput and low operational cost due to shared infrastructure, they rely on external audit mechanisms and are susceptible to unauthorized tampering. In contrast, blockchain-based logging systems offer built-in audit trails through cryptographic hash chains and immutable records, ensuring tamper resistance and traceability, albeit at a reduced throughput and moderate gas costs. As summarized in Table 1, blockchain's native alignment with GDPR/HIPAA data integrity principles positions it as a viable candidate for medico-legal scenarios requiring high trust and verifiability. Table 1 presents the comparative study between centralized and blockchain logging.

1.2. Motivation

Diabetes is a growing global health concern, leading to severe complications and a high economic burden. Research in this field is essential for early detection, personalized treatment, and smart healthcare solutions. AI-driven models can predict diabetes risks, while IoT-based wearable sensors enable real-time monitoring. Blockchain ensures secure data management, and telemedicine improves access to care, especially in remote areas. Advancements in non-invasive glucose monitoring, smart insulin therapy, and AI-powered drug optimization can revolutionize diabetes management. By integrating AI, IoT, and blockchain, researchers can develop more effective, secure, and patient-friendly solutions to improve diabetes care worldwide.

1.3. Contribution

The following are the major contributions of our proposed work in diabetes prediction and healthcare security:

1. Proposed a hybrid technique: Stacking of Machine Learning Classifier with Deep Learning Classifier

- A hybrid approach is introduced by combining **machine learning (ML)** classifiers with **deep learning (DL)** models using a stacked framework.
- This technique enhances **generalization and predictive accuracy** by leveraging both structured feature learning from ML and complex pattern recognition from DL.
- The stacked model effectively captures **non-linear relationships and hidden patterns** in diabetes datasets, improving prediction outcomes.

2. Higher Accuracy Compared to Existing Work on the PIMA Diabetes Dataset

- Our stacked ML-DL model outperforms traditional classifiers (e.g., Logistic Regression, Decision Trees, SVM) and even standalone deep learning models.
- Hyperparameter tuning and advanced feature selection methods contribute to significantly higher accuracy compared to previous research.
- Comparative analysis shows that our model effectively reduces **false positives and false negatives**, leading to more reliable diabetes predictions.

3. Integration of Blockchain with Machine Learning

- The system integrates **blockchain technology** with AI for secure, transparent, and immutable healthcare data storage and access.
- Blockchain ensures **tamper-proof medical records**, allowing only authorized healthcare professionals to access patient data.
- The use of **smart contracts** enforces access control policies, ensuring privacy and preventing unauthorized modifications.
- This integration significantly enhances **data security, patient privacy, and the integrity of medical records**.

4. Early Diabetes Prediction System

- Early detection of diabetes is crucial in preventing severe complications such as heart disease, kidney failure, and neuropathy.
- Our model can use **historical health data, real-time sensor readings (e.g., glucose levels, BMI, blood pressure)**, and lifestyle factors for early risk assessment.
- AI-driven predictive analysis helps healthcare providers to suggest **preventive measures, lifestyle modifications, and timely interventions**.
- The system can be integrated into **IoT-based wearable healthcare devices** for continuous monitoring and proactive health alerts.

1.4. Paper organization

The subsequent sections of the paper are organized as follows. Section 2 discusses the pertinent works and recommendations proposed by various authors. Section 3 presents the proposed framework. Section 4 delineates the materials and technique employed in this study. Section 5 presents the findings and a discourse on the latest works. Section 6 presents a summary of the argument, concludes, and proposes further directions. The limitations of the proposed work are presented in Section 7.

2. Related works

This section presents a survey of various research papers. Many researchers and scholars have used the potential of ML techniques for prediction and analysis accurately and keeping in mind the security and privacy of user data.

2.1. Machine learning in diabetes detection

Data sets used by Nazin Ahmed et al. [6] were pre-processed using label encoding, missing value handling, and outlier elimination. They used several machine-learning methods. Study data was preprocessed and seven ML models were used. Models included NB, DT, RF, SVM, LR, XGBoost, and KNN. SVM attained the highest accuracy rate of 80.26%. Talha Mahboob et al. [7] employed attribute selection and association rule mining in their investigation. An accuracy of 75.7% was achieved using ANN. Scholars like Salliah Shafi et al. [8] and Jobeda Jamal Khanam [9] et al. utilized feature engineering and K-Fold cross-validation to assess projected data accuracy. Shafi et al. [8] achieved 74.28% accuracy with SVM, DT, and NB on the PID dataset, whereas Khanam et al. [9] achieved 79.42% accuracy with KNN, DT, RF, NB, AB, LR, and SVM algorithms. Majumdar et al. [10] employed pipeline algorithm to forecast values. AdaBoost achieved the greatest accuracy of 98.8%. SVM, DT, AdaBoost, Stacking, and RF algorithms were employed by Namrata Singh et al. [11] to obtain an accuracy of 83.8% utilizing NSGA-II stacking.

M.A.R. Refat et al. [12] employed SVM, DT, AdaBoost, Stacking, and RF ensemble machine learning techniques. It reached a maximum accuracy of 83.8% utilizing NSGA-II stacking. Souad L.M. Sainte et al. [13] used all recent machine learning algorithms. Implemented machine learning classifiers with 0 or 1 frequency on the Pima Indian diabetes dataset, achieving 74.48% accuracy. Sarwar et al. [14] used NB, KNN, SVM, LR, DT, and RF algorithms to predict disease on the PID dataset. For this investigation, the dataset was separated into testing and training sections. Testing made up 30% of the dataset, while training made up 70%. The six algorithms were applied to Enthought Canopy for results. The SVM and K-NN models have the maximum accuracy of 77%. The paper [15] examines ML approaches for diabetes prediction using PIMA. Correlation-based feature selection and AdaBoost classification were used. Along with MLP, SVM, and LR stacking, a new method was devised. The proposed stacking method outperforms AdaBoost and other classifiers in predicted accuracy. The model also performed well in the Cleveland Heart Disease and Wisconsin Breast Cancer datasets, demonstrating broad application. Overall accuracy was 78.2% in the PIMA data set.

A robust pipeline-based ensemble machine learning framework for multi-class diabetes prediction was proposed by Abnoosian et al. [16] using the Iraqi Patient Dataset for Diabetes (IPDD) from Al-Kindy Teaching Hospital, Iraq The study used the One-vs-One approach to merge K-NN, SVM, Decision Tree, Random Forest, AdaBoost, and Gaussian Naive Bayes into a weighted AuC-based ensemble. Advanced preprocessing (k-NN missing value imputation, normalization, MRMR feature selection, and PCA/ICA dimensionality reduction) and grid search and Bayesian optimization for hyperparameter tuning are key breakthroughs. The ensemble model (K-NN + AB + DT + RF) outperformed existing models with 99.87% accuracy and 0.999 AuC, providing a solid method for early diabetes detection. Abnoosian et al. [17] developed a pipeline-based classification framework for diabetes prediction using two datasets: Indian Diabetic Patient Dataset (binary classes) and Iraqi Patient Dataset (healthy, prediabetic, and diabetic classes) from regional health records. Advanced preprocessing, feature selection, One-vs-One classification, and hyperparameter optimization address class imbalance and increase prediction performance. ML models were tested, with AdaBoost attaining 89.98% accuracy and 94.11% AuC on the Indian dataset and Random Forest earning 98.66% accuracy and 98.62% AuC on the Iraqi dataset. The study shows the model's clinical decision support system integration potential.

Farnoosh et al. [18] developed a pipeline-based machine learning framework to predict COVID-19 patient survival and mortality using the COVID-19 Mexican Patient Dataset (MPD) from public health records. To reduce dimensionality and improve prediction, the study used multiple ML models with hyperparameter optimization and feature selection approaches such MRMR, permutation feature importance, PCA, and ICA. Use of the KNN algorithm with four principal components yielded an outstanding AuC of 100%, indicating promise for clinical decision support in COVID-19 management. Zhou et al. [19] proposed a machine learning framework combining PIMA and BD data sets. Model generalizability was assessed via intra-, inter-, and partial fusion validation. In PIMA, XGBoost had 79% accuracy, whereas RF and GB reached 99% on BD. Using PIMA training and BD testing, an ensemble model scored 88%, but plummeted to 74% when reversed. Partial fusion increased DL model accuracy to 98%, highlighting the importance of data diversity.

2.2. Blockchain in diabetes detection

M. Chen et al. proposed a method for diabetes diagnosis utilizing Blockchain technology [20]. The system comprises three stages: (i) registration, (ii) user verification, and (iii) uploading IoT data utilizing Blockchain technology. The user verified their identity through the authentication unit utilizing the Electronic Health Records Manager. Machine learning algorithms enabled the detection of a patient's diabetes and its secure sharing with their healthcare provider. The physician could convey the results to the patient due to machine learning methodologies. Several scholars have utilized a conceptual insurance model grounded in smart contracts. A. Karmakar and colleagues [21] proposed ChainSure, a conceptual insurance framework utilizing smart contracts. It had been created and tested on the Ethereum test network. Their objective was to achieve complete automation of the system, with blockchain technology serving as the pivotal element. Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS). CPA is accessible to all stakeholders through the Internet once it is deployed on the ChainSure cloud. A framework developed by A. Gupta et al. [22] consists of: (a) registration authority, (b) smart contract, (c) interplanetary file system, and (d) decision-making. These principal entities, comprising medical service providers and patients, register with the authority to facilitate secure contact with a blockchain network. The user verifies their identity by referencing the EHRs Manager data stored in an authentication unit.

Machine learning techniques facilitated the identification of diabetes in patients and the secure dissemination of results to healthcare professionals. The Summary of the above works is presented in Table 2.

3. Proposed framework

The proposed system ensures secure healthcare data analysis and storage using blockchain and IPFS. It starts with patient and doctor registration, where details are stored securely. Next, patient health data is collected from sensors like glucometers and BP sensors. The collected data is encrypted and stored securely, with hashes recorded on the blockchain and raw data stored in IPFS. Smart contracts manage access requests, ensuring only authorized healthcare providers retrieve data. Providers analyze the data and use stacked machine learning and deep learning models to predict potential health risks. Finally, analysis results are securely stored and retrieved while maintaining privacy through smart contracts. The algorithm 1 presents complete work flow of our proposed work i.e., StaBloCare, a Blockchain enabled medical and healthcare system for diabetes detection in early the stages. Fig. 2 demonstrates the proposed architecture and its working with all the communication links between all the different entities. The structured workflow of the proposed system is categorized into the following phases:

- **User Registration:** The initial phase focuses on user registration, encompassing both patients and healthcare providers. During this process, users submit their personal information, which is securely stored by the registration authority. Furthermore, a unique cryptographic key pair is generated for each user, ensuring their authentication within the system.

Table 2
Comparison of Different Techniques for Diabetes Prediction.

Author	Dataset	Techniques Used	Accuracy (%)
Khanam et al. [9]	PIDD	DT, KNN, RF, NB, AB, LR, SVM	79.42
Talha Mahboob Alam et al. [7]	PIDD	ANN, K-Means, RF	75.70
M. A. Sarwar et al. [14]	PIDD	KNN, NB, SVM, DT, LR, RF	77.00
Salliah Shafi et al. [8]	Clinical Data, Bandipora Hospital	RF, LR, DT, Gradient Boost, SVM, MLP	98.00
Salliah Shafi et al. [8]	PIDD	NB, SVM, DT	74.28
Aishwarya Majumdar et al. [10]	Diabetes Dataset	AB, GB, ETC, LDA	98.80
Souad L.M. Sainte et al. [13]	PIDD	REPTree, M5P, PART	74.40
Namrata Singh et al. [11]	PIDD	NSGA-II Stacking, SVM, REP, RF	83.80
Nazin Ahmed et al. [6]	PIDD	SVM, NB, DT, RF, LR, GB, KNN	80.26
Deepti Sisodia et al. [23]	PIDD	NB, SVM, DT	76.30
M. Chen et al. [20]	PIDD	DT, KNN, RF, LR, SVM	80.51
N. Sneha et al. [24]	UCI ML	SVM, RF, NB, DT, KNN	77.30
S.M. Ganie et al. [25]	Diabetes Dataset4	RF, LR, NB, SVM, DT, ANN	93.79
S. Kumari et al. [26]	PIDD	Soft Voting, LR, KNN, SVM, AB, Bagging	79.08
R. D. Joshi et al. [27]	PIDD	LR, Classification Tree	78.26
Refat et al. [12]	Sylhet Clinical Dataset	XGB, RF, DT, KNN, SVM, ANN, MLP, LSTM	≈100
H. Lu et al. [28]	CBHS Health Funds	LR, KNN, SVM, NB, DT, RF, XGB, RNN	84.95
Quan Zou et al. [29]	PIDD	DT, RF, NN	77.21
Md. Maniruzzaman et al. [30]	NHANES	NB, DT, AB, RF	94.25

- **User Authentication:** This phase is critical to protecting blockchain and IPFS data security and their privacy. It entails checking user's identities and providing or refusing system access. The solution prevents data breaches and unauthorized access using strong authentication.
- **Data Analysis and Model Learning:** After successful authentication, the system transitions to data analysis and model learning. At this stage, health-related data collected from users is processed using advanced algorithms and machine learning techniques. This analysis facilitates valuable insights, empowering healthcare professionals to make informed decisions and provide personalized medical care.
- **Blockchain Integration for Data Integrity:** After the previous phase, IPFS stores its hash string on the blockchain. This phase guarantees a persistent, tamper-proof record of IPFS data, boosting system security. In this step, the smart contract securely saves encrypted transaction data from verified users and analyses outcomes on IPFS. IPFS creates a secure identity and access key for stored data with a unique hash string. This safeguards data and prevents unauthorized changes.
- **Blockchain Integration for Data Integrity:** Following the previous phase, the hash string generated by IPFS is stored in the blockchain. This step ensures a permanent and tamper-proof record of the data stored on IPFS, reinforcing the security and integrity of the system.

3.1. Machine learning framework

We employ several machine learning classifiers which are stacked and given to deep learning classifier for diabetes prediction. The stacking technique of machine learning classifier with deep learning model improves the classification accuracy of proposed model to a large extent. Different classifiers used in the proposed work are discussed below.

3.1.1. K-Nearest Neighbors (KNN)

A data point's kth nearest neighbor is found using the distance metric in K-Nearest Neighbor [31]. KNN, k-NN, and other acronyms refer to the k-nearest neighbors method, a supervised learning classifier. K-NN is usually used for classification, but it can also handle regression. KNN is slow in learning from training data, thus it retains it and executes the action directly when testing. As a non-parametric approach, K-NN makes no assumptions about the data.

3.1.2. Decision tree

The decision tree [32] is a widely utilized machine learning algorithm that is applicable to both classification and regression tasks. It operates by recursively splitting the input data into subsets based on the values of the input feature, constructing a tree that represents a succession of binary decisions. At each decision juncture, the algorithm selects the feature that yields the greatest information gain or diminishes ambiguity regarding the target variable. The process continues until the data are completely partitioned or the stopping criteria are satisfied. A decision tree is a classifier represented as a recursive partition of the instance space.

3.1.3. Random forest

A random forest is an analytical machine learning technique that constructs a network of decision trees to provide predictions [33]. The algorithm uses a series of decision trees trained on subsets of the input data and features to arrive at a final prediction, which is then combined. When training a decision tree, it is common practice to randomly select some input data and some features per each tree split. The model's accuracy and robustness against overfitting are both improved by this randomization. The average of the forecasts made by each decision tree in the forest is used to arrive at the final forecast. More accurate predictions are produced by this ensemble method, which successfully decreases the model's variance.

3.1.4. Naïve Bayes

When it comes to statistics, Bayesian classifiers are the way to go. For example, they can estimate the likelihood that a specific sample is a member of a certain class. Naïve Bayes classifiers can quickly classify fresh data points and are easy to train [34]. They also show good resistance to overfitting, a prevalent issue in machine learning. Bayes' theorem is the foundation of a Bayesian classifier. Naïve Bayesian classifiers presume that the impact of a single attribute value on a specific class is unrelated to the values of any other attributes. Class conditional independence is the name given to this assumption [35].

3.1.5. XGBoost

XGBoost (Extreme Gradient Boosting) method was proposed by Tianqi Chen and Carlos Guestrin for the first time in 2011. Since then, it has been optimized on a regular and continuous basis by the research scholars as well as scientists. The XGBoost library was developed with the goal of becoming a highly efficient, versatile, and portable distributed gradient boosting tool [36]. Machine learning methods are applied within the framework of Gradient Boosting. For a wide variety of data science issues, XGBoost offers a parallel tree boosting (GBDT, GBM) that works quickly and accurately. The code is compatible with Hadoop, SGE, and MPI, three of the most prominent distributed environments, and it is capable of solving problems with billions of samples or more.

3.1.6. Support vector machine

As a subset of supervised learning, SVMs are machine learning classifier. Li et al. [37] used SVM for regression and classification challenges. This method generates an n-dimensional space categorization decision border to detect new data points. Stein and Li named these decision boundaries hyperplanes. Support vectors are data points near to the hyperplane that affect its direction and placement. Support vectors are used to increase classifier margin. Support vector removal creates hyperplane location to change.

3.1.7. Logistic regression

Its goal is to forecast which of two categories a given input falls into. Uses a logistic function to estimate the input's likelihood of belonging to a specific class based on its attributes. While both linear regression and logistic regression are linear models, logistic regression [38] differs in that it employs a sigmoid activation function to convert the linear output into a probability value ranging from 0 to 1, distinct from linear regression. Logistic classifiers often use a line or hyperplane as their decision boundary, which splits the input space into two areas representing the two classes.

3.1.8. Stacking classifier

In stacking [39] technique, we have combined all seven machine learning classifiers that is SVM, Naive Bayes, Random forest, XGBoost, KNN, Decision tree and logistic regression to make base classifier. This base classifier is fed to deep learning model for final prediction. The architecture and configuration of meta deep learning classifier are presented in Tables 3 and 4 respectively. Table 3 outlines the deep learning meta-learner architecture in the proposed stacking framework. It has four layers: a input layer consisted of 7 neurons corresponding to the 7 predictions done by the 7 base classifiers from the PIMA dataset, a dense layer with 64 neurons and ReLU activation, a dropout layer with 30% regularization, a second dense layer with 32 neurons and ReLU activation, and a final output layer with a single neuron with sigmoid activation for binary classification. The diabetes prediction model optimizes learning capacity and computing efficiency with 2,625 trainable parameters. The flowchart of the stacked deep learning classifier is shown in Fig. 1. Table 4 shows the deep learning-based meta-learner training configuration. The model is optimized with Adam and trained with 8 batches over 50 epochs. The binary classification job uses binary cross-entropy as the loss function, and accuracy is the main performance parameter during training. The meta-classifier in a stacked model consists of two stages, base classifiers and the meta-learner (deep learning model) which works as follows:

A. Base classifiers (ML models)

Let X be the input feature matrix with N samples and M features:

$$X = [x_1, x_2, \dots, x_N] \in \mathbb{R}^{N\times M} \tag{1}$$

Each base classifier $f_i(X)$ produces an output prediction:

$$h_1 = f_1(X), \quad h_2 = f_2(X), \quad \dots, \quad h_j = f_j(X) \tag{2}$$

where j is the number of base classifiers.

Thus, the meta-training dataset becomes:

$$H = [h_1, h_2, \dots, h_j] \in \mathbb{R}^{N\times j} \tag{3}$$

where each h_i is a prediction vector from a base model.

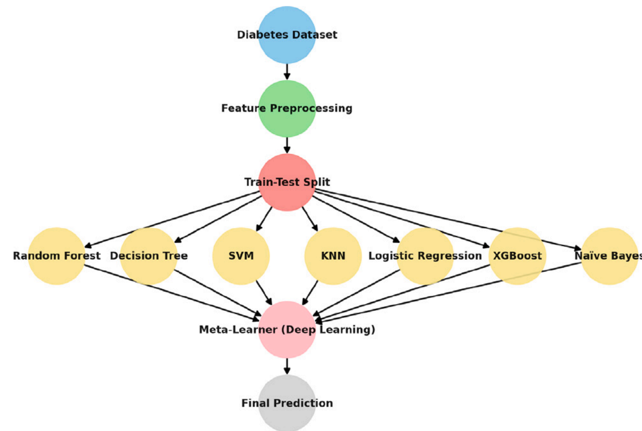


Fig. 1. Flowchart of stacking machine learning classifiers into deep learning model.

B. Meta-learner (deep learning model)

The meta-learner takes H as input and learns a final mapping:

$$\hat{S} = g(H) = \sigma(WH + b) \tag{4}$$

where:

- The weight matrix and bias for the deep learning model are represented by W and b, respectively.,
- The activation function, such as ReLU or sigmoid, is represented by σ,
- $\hat{S}$ is the final predicted outcome.

The ultimate goal is to reduce the loss of binary cross-entropy as little as possible:

$$l = -\sum_{i=1}^{N} \left[S_i \log \hat{S}_i + (1 - S_i)\log(1 - \hat{S}_i)\right] \tag{5}$$

where S_i is the true label.

Theoretical analysis: Invariance property of stacking

Stacking, or stacked generalization, is an ensemble learning technique where the predictions of multiple base learners are used as input features to a higher-level meta-learner. Formally, let $\mathcal{D} = \{(\mathbf{X}_i, S_i)\}_{i=1}^N$ be a dataset with input vectors $\mathbf{X}_i \in \mathbb{R}^d$ and corresponding labels S_i. Suppose we have j base classifiers $f_1, f_2, \dots, f_j$ trained on $\mathcal{D}$, and a meta-learner G trained on their outputs.

$$\mathbf{z}_i = \left[f_1(\mathbf{X}_i), f_2(\mathbf{X}_i), \dots, f_j(\mathbf{X}_i)\right] \tag{6}$$

$$\hat{S}_i = G(\mathbf{z}_i) = G\left(f_1(\mathbf{X}_i), f_2(\mathbf{X}_i), \dots, f_j(\mathbf{X}_i)\right) \tag{7}$$

Now, consider a permutation π of the base classifiers. The reordered prediction vector becomes:

$$\mathbf{z}_i^\pi = \left[f_{\pi(1)}(\mathbf{X}_i), f_{\pi(2)}(\mathbf{X}_i), \dots, f_{\pi(j)}(\mathbf{X}_i)\right] \tag{8}$$

If the meta-learner G is permutation-invariant (i.e., $G(\mathbf{z}_i) = G(\mathbf{z}_i^\pi)$), then the final prediction is invariant under the order of base learners. However, most practical meta-learners such as logistic regression or neural networks are sensitive to input order. Therefore, reordering the base learners without retraining G may result in different predictions:

$$G(\mathbf{z}_i) \neq G(\mathbf{z}_i^\pi) \tag{9}$$

Table 3
Deep Learning Meta-Learner Architecture.

Layer	Type	Input Size	Output Size	Activation	Trainable Parameters	Description
1	Dense	7	64	ReLU	$(7+1)\times 64 = 512$	Includes weights and bias
2	Dropout	64	64	–	0	30% units dropped during training for regularization
3	Dense	64	32	ReLU	$(64+1)\times 32 = 2080$	Fully connected hidden layer
4	Dense	32	1	Sigmoid	$(32+1)\times 1 = 33$	Binary classification output
Total Trainable Parameters					**2625**	

Table 4
Meta-Learner Training Configuration.

Configuration	Value
Evaluation Metric	Accuracy
Optimizer	Adam
Batch Size	8
Loss Function	Binary Cross-entropy
No. of Epochs	50

Hence, **the stacking algorithm is not invariant to the order of base learners** unless the meta-learner is designed to be permutation-invariant or is retrained for each new order. This theoretical observation is also supported by empirical results, which show accuracy variations when base learner order is changed.

3.2. Blockchain framework

In our research, we considered a public blockchain [40] setting akin to Ethereum. Patients and doctors can use Ethereum's features to communicate with a smart contract because we have designated them as users in our blockchain system. Both patients and healthcare personnel must register themselves as an Externally Owned Accounts (EOAs) under a smart contract to be recognized as users. EOAs are private key controlled Ethereum accounts that are used to sign and approve transactions that are sent over the network. During registration as an Externally Owned Account (EOA), each entity generates its unique Ethereum address (EA). EA that serves as their user identity. Although the EA is accessible to everyone, it does not reveal the user's identity. Users can use their private keys to transmit transactions to the smart contract to invoke functions after registering. Users can securely engage with the system since the smart contract processes these transactions and carries out the associated tasks. With this strategy, access to healthcare information and services is restricted to authorized users only.

The Contract Account (CA) of a smart contract serves as a unique identifier. The contract owner creates it, outlining the terms and capabilities of the agreement. After being formed, a transaction is sent to deploy the smart contract onto the Ethereum blockchain, giving it an address. Each Ethereum node, which is a computer running the Ethereum software and helping to maintain the network, stores a copy of the contract. The smart contract's capabilities are carried out in a safe and deterministic manner via the Ethereum Virtual Machine (EVM). The blockchain records every action taken by the smart contract, ensuring transparency and audit-ability. The blockchain acts as a publicly accessible, immutable ledger for transactions. This offers an automatic method that is transparent, lowering the possibility of mistakes or manipulation. Block validation also makes sure that newly generated blocks adhere to the Ethereum blockchain's rules and regulations. Verifying transactions, consensus guidelines, and miner authorization are all part of the validation process. Instead of the conventional Proof of Work (PoW) technique used in Ethereum, we have chosen the Proof of Stake (PoS) approach. With PoS, validators must stake their own coin as security to ratify transactions. Based on their stake, validators are chosen at random, which encourages them to be trustworthy. This makes it more appropriate for applications that call for high throughput, low latency, and cheap transaction costs by reducing computing complexity and energy usage compared to PoW.

Algorithm 1 Blockchain-Based Healthcare Data Processing with Stacked ML and DL

Input: $P_{details}$ (Patient Registration Details), $D_{details}$ (Doctor Registration Details), S_{data} (Sensor Data)
Output: Secure Healthcare Data Analysis and Storage
Step 1: Patient and Doctor Registration
$R_A \leftarrow Register(P_{details}, D_{details})$ ▷ Register with Registration Authority
Store $P_{details}, D_{details}$ in **Patient Record** and **Doctor Record**
Step 2: Collect Patient Health Data
for each patient P_i **do**
    $S_i \leftarrow \{Glucometer, BPSensor, SkinFold, BMI\}$ ▷ Collect Sensor Data
end for
Step 3: Secure Data Storage Using Blockchain and IPFS
for each patient P_i **do**
    Encrypt S_i and store hash H_i on the **Blockchain**
    Store S_i in **IPFS (InterPlanetary File System)**
    Link H_i to S_i for verification
end for
Step 4: Smart Contracts for Data Access
Deploy **Smart Contracts** to:
for each request R_j from a healthcare provider **do**
    if Valid Access Request **then**
        Approve access and retrieve H_i
    else
        Deny access
    end if
end for
Step 5: Healthcare Service Provider Analysis
Healthcare Provider retrieves and processes S_i
Perform Data Analysis and Risk Assessment
Step 6: Stacked Machine Learning with Deep Learning-Based Diagnosis
Train **Stacked ML and DL Model** using historical patient data S
Predict potential health risks: $\hat{y} = Model(S)$
Generate **Data Analysis Results**
Step 7: Store and Retrieve Analysis Results
Encrypt and store results securely on Blockchain and IPFS
Allow retrieval of processed data using hashes
Enforce privacy-preserving access control through Smart Contracts
End Algorithm

3.2.1. Registration authority

The registration authority collects and stores basic hospital staff information, including doctors, surgeons, and nurses. A user can only register for a smart contract once, and their duties vary per feature. Smart contracts generate unique EAs for each user. After smart contracts are registered, the registration authority stores their data for future reference. Registration authorities give each entry a unique ID. In addition to the user ID, the registration authority collects other data. This data includes patient's names, addresses, gender, age, and health service provider's names, qualifications, addresses, etc. The algorithm 2 shows the user's process step-by-step.

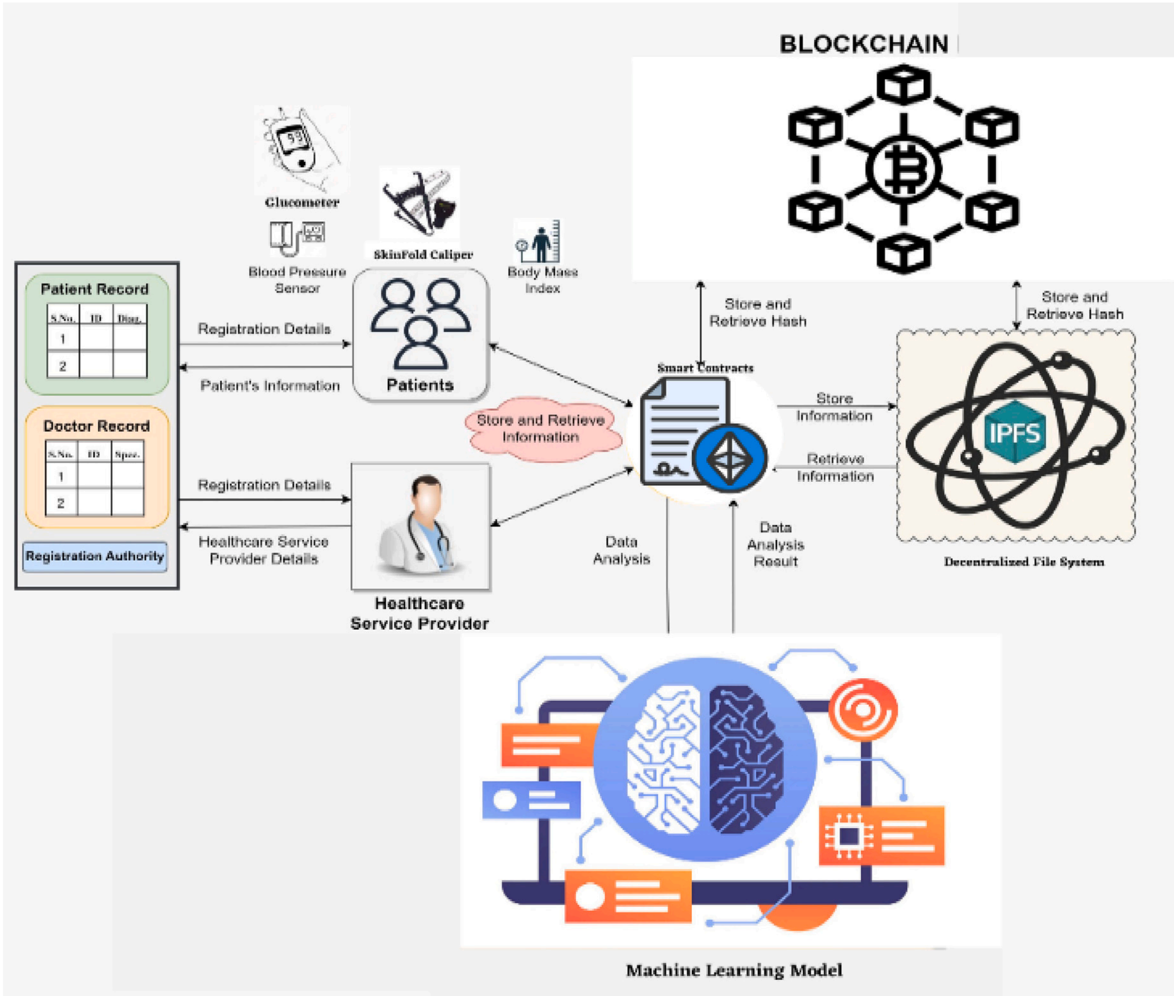


Fig. 2. Blockchain and Machine Learning in Healthcare System.

Algorithm 2 Algorithm for User Registration

1: **begin**
2: **if** ($u.EA$ is valid) **then**
3: User is already registered.
4: **else**
5: Register user to smart contract and generate corresponding $u.EPU$ and $u.EPK$.
6: Assigns the role of either patient or health service provider.
7: **if** role = patient **then**
8: Add an entry in the patient record with identity, public key, private key, name, address, gender, age, etc.
9: **else**
10: Add an entry in the healthcare service provider record with its public key, private key, name, qualification, contact, address, etc.
11: **end if**
12: **end if**
13: **end**

3.2.2. Smart contract

Smart contracts are key to the proposed model. It decentralizes network management by managing peer functioning independently. It prevents centralized controller issues like single point of failure. Smart contracts, like programming language contracts, include predefined features that trigger when certain conditions occur. Smart contracts are often referred to as "autonomous agents" because of their independence from individual users or organizations. Any Ethereum node can activate a smart contract using an Externally Owned Account (EOA), a public–private key pair linked to a user. Smart contracts can be activated by messages from other smart contracts, enabling sophisticated network interactions. The blockchain-enabled architecture's public channel makes user communication vulnerable to interceptions and changes. The framework uses public–private key authentication to address this issue. User data can be stored and retrieved via a unique public–private key pair. Users must authenticate to the smart contract using their private key to access decentralized storage. If authentication is successful, the smart contract allows storage access based on duties. If authentication fails, the smart contract will reject the user's request and prevent future contact. This helps restrict access to distributed storage and prevent unauthorized users from doing so.

The proposed architecture uses a public–private key structure and authentication to provide secure access to decentralized storage without compromising user data. This ensures that only approved users utilize public communication channels and the system. The user authentication process begins with $u.T_{auth}$ token creation. A smart contract generates a unique authentication for each person accessing data. The following formula and current time stamp are used to calculate $u.T_{auth}$ from user-related data: The user tells the smart contract to receive the encrypted communication. This encryption uses the user's private key: After receiving the message, the smart contract decrypts it using the user's public key to access the data. The smart contract then checks if the request came from an existing user. A smart contract refuses the request and stops all further actions. If not, it verifies the data further. The smart contract generates $u.T'_{auth}$ again using the same data and hashing method. The smart contract authorizes the requesting user if $u.T_{auth} = u.T'_{auth}$. Step-by-step user authentication utilizing smart contract and registration authority is illustrated in Algorithm 3. $O(n)$ is computational complexity for user authentication.

Algorithm 3 Algorithm for User Authentication at Smart Contract

begin
Decrypt the message using the user's public key.
$D_{u.EPU}(E_{u.EPK}(u.T_{auth}||data||h(data)||u.EA||TS))$
if ($u.EA$ is valid) **then**
  Generate the authentication token $u.T'_{auth}$.
  if ($u.T_{auth} = u.T'_{auth}$) **then**
    User authenticated successfully.
  else
    Authentication of user u is unsuccessful.
  end if
else
  Unregistered user.
end if
end

3.2.3. Interplanetary file system (IPFS)

Support for IPFS and Ethereum-based smart contracts encourages the proposed framework to use IPFS. The conventional central cloud storage system has these drawbacks. Centralized storage systems have more single points of failure. Distributed storage solutions improve data portability, flexibility, and reduce data loss. IPFS handles large files well. Due to their limited storage capacity, blockchain-based storage solutions may struggle to store large data. IPFS solves this by splitting large files into digestible chunks and distributing them over the network. This approach efficiently stores and accesses large files, preventing any network node from exceeding its storage capacity. IPFS addresses content rather than memory locations, unlike memory location-based storage solutions. Each IPFS file has a unique cryptographic hash value that serves as its address. This approach removes duplicate items since identical files have the same hash value. It also makes it easy for users to verify file accuracy because any file changes change the hash value. IPFS also stores each file's history, allowing users to access prior versions. Smart contracts provide secure and verified patient-hospital image data exchange in the proposed task. IPFS distributed hashes and encrypted picture data are stored on the immutable blockchain network. Hash value is used to access IPFS data. Only registered and permitted users can access saved data via the smart contract. Once properly saved, IPFS returns a unique cryptographic hash, making data changes difficult. Blockchains store hashes to secure and simplify data access.

3.2.4. Security analysis

To ensure data privacy and accuracy, the suggested diabetes disease healthcare monitoring system detects several security precautions. Blockchain hash values store and validate files for security. One-way functions produce the original hash values, making computational retrieval impossible. This prevents secret blockchain data modifications. Smart contracts and public–private key cryptography restrict decentralized storage system access to authorized users. Blockchain data can only be restored by authorized individuals with the relevant private key. Role-based data access restricts improper access. This protects patient data and limits authorized users. Before sending data to IPFS, the suggested method encrypts it. An attacker needs the decryption key to read or alter data.

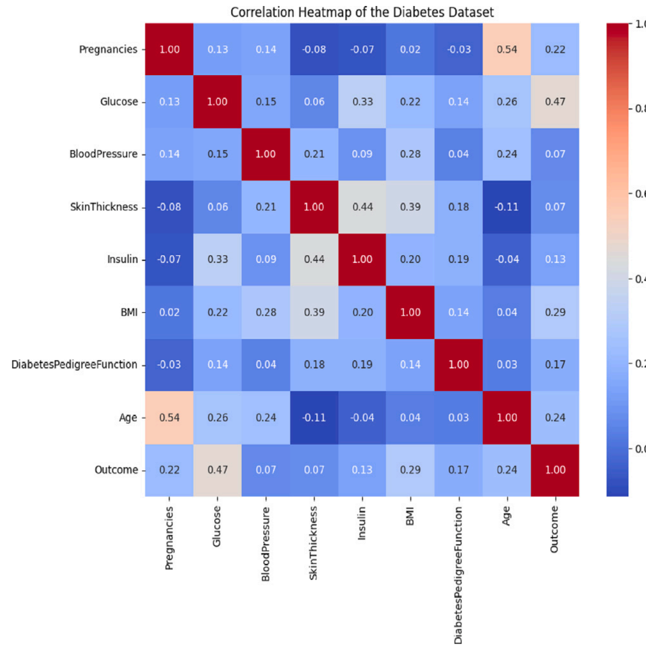


Fig. 3. Correlation heatmap of the diabetes dataset.

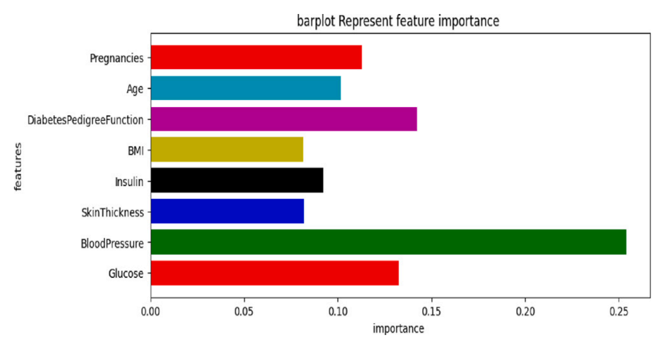


Fig. 4. Bar plot representing Feature importance.

4. Materials and methodology

In our study, we utilized the Pima Indian Diabetes Database dataset [41], which was acquired from the University of California, to provide diabetes predictions. There are 768 rows and 9 columns in this dataset. The description of the attributes of the dataset is presented in Table 5. The correlation heatmap of the diabetes dataset is presented in Fig. 3.

The importance of the feature for diabetes prediction is shown in Fig. 4. From the figure it is clear that blood pressure is most important feature in determining diabetes. Other important features include diabetes pedigree function, glucose, pregnancies etc.

The attributes are scaled using Standard-scaler so that no one attribute can overshadow the others in the forecasts. The dataset is thereafter divided into 80% for training and 20% for testing in order to evaluate the model. The dataset facilitates the training of many classifiers, such as Naïve Bayes, Decision Tree, K-Nearest Neighbors, Support Vector Machine, XGBoost, and Random Forest. Various important performance indicators are used to assess the models, including Accuracy, F1 Score, Precision, Recall, and AUC Score. To further analyze the data and model predictions, visualization techniques such as bar graphs, scatter plots, and histograms are used to extract insights and understand data distributions effectively. Table 6 presents a discussion of the measures used to gauge the success of our proposed model:

Table 5
Description of PIMA Indian Diabetes Dataset Features.

Feature	Description
Pregnancies	Number of times a woman has been pregnant
Glucose	Plasma glucose concentration after 2 h in an oral glucose tolerance test
Blood pressure	Diastolic blood pressure (mm Hg)
Skin thickness	Triceps skin fold thickness (mm)
Insulin	2-hour serum insulin (μU/ml)
BMI	Body mass index (weight in kg/height in m^2)
Age	Age (years)
Diabetes pedigree function	Determines the risk of diabetes by analyzing family medical records
Outcome	Target variable: 1 (has diabetes), 0 (does not have diabetes)

Table 6
Common classifier's performance evaluation metrics used in this article.

Performance metric	Description	Formula
Accuracy	It is a tool for evaluating a model's efficacy. An anomaly occurs if there is a misalignment of classes.	$\frac{P+Q}{(P+S)+(R+Q)}$
Precision	As soon as false positives are not tolerated, precision metrics become appropriate. One way to measure accuracy is by comparing total positive predictions to total P.	$\frac{P}{(P+R)}$
Sensitivity(Recall)	How many false positives out of a total number of positives is the sensitivity.	$\frac{P}{P+S}$
Specificity	Specificity is the right metric to use when you want to include all genuine negatives. Q divided by the sum of all negatives is one way to put it.	$\frac{Q}{Q+R}$
F1-Score	It can be calculated as	$\frac{2*P}{(2*P+S+R)}$
AuC- RoC	When determining the efficacy of a model on a well-balanced dataset, AUC-ROC is employed.	
FPR	False positive rate is what it stands for. A competent classifier will have a value close to zero.	$\frac{R}{R+Q}$
FNR	It is short for the rate of false negatives. A competent classifier will have a value close to zero.	$\frac{S}{P+S}$
MCC	A high score is only given if the prediction was correct in all four regions of the confusion matrix, making it a more trustworthy statistical rate.	$\frac{P*Q-R*S}{\sqrt{(P+R)*(P+S)*(Q+R)*(Q+S)}}$

P = Rightly classified samples. **Q** = Normal samples classified rightly.
R = Normal samples classified as Abnormal. **S** = Abnormal samples classified as Normal.

Confusion Matrix: True positives, true negatives, false positives, and false negatives are all represented by different rows and columns in a confusion matrix.

5. Experimental results

The diabetes data set containing 768 records and 9 attributes was collected to identify patterns and trends that could help predict the value. The data was standardized to increase the accuracy of the classification algorithms. Data was then split into 80% training data and 20% test data.

5.1. Results based on stacking DL method

We used various Machine Learning algorithms to train and test the data. The Table 7, shows the comparison of the accuracies of these algorithms. RF method recorded an accuracy of 87%. NB, LR, SVM, DT, KNN, and XGBoost algorithms achieved an accuracy of 70.5%, 75%, 75%, 80%, 80.5%, and 83.5% respectively. Stacked machine learning based deep learning method outperformed all the other algorithms used. Stacking classifier obtained an accuracy of 90.5%.

The ROC is a curve that displays the classification model's performance over all thresholds. At various categorization thresholds, a ROC curve compares TPR and FPR. Both True Positives and False Positives are increased when the classification criterion is lowered because more objects are classified as positive. Fig. 5 shows the comparison of accuracy for various classifiers used. Fig. 4 shows the level of dominance of different attribute in diabetes. It should be noted that glucose is most important factor and thereafter BMI and Age are other two important factors. Fig. 7 presents Receiver operating characteristics for different classifier used here. From the graph, we can see that Random forest and stacked classifier have higher area under curve value. This suggests that these two classifier is well suited for proposed model. The Fig. 7 shows a typical ROC curve. The AuC value of Random Forest and stacking classifier in our study was 92% and 92.11%. Naïve Bayes, logistic regression and KNN had achieved AuC value of 82.8%, 86% and 87.1% respectively. Decision Tree, SVM and XGBoost achieved an AuC value of 80.1%, 86.3% and 90.5% respectively.

The Table 8 illustrates the impact of different base classifier arrangements on the ultimate model accuracy inside a stacking ensemble framework. Seven permutations of classifiers — RF, SVM, LR, NB XGBoost, KNN, and DT — were assessed. The accuracies obtained varied from 0.88 to 0.905, with the maximum accuracy of 0.905 attained through several configurations (e.g., indices 3, 4, and 6). The results indicate that stacking enhances predictive performance, whereas the sequence of classifiers may have a minor, albeit insignificant, effect, hence corroborating the partial invariance property of stacking ensembles. Also, we have taken three different activation functions to prove the consistency of the outcome and presented the result in Table 9 before claiming the accuracy. This is also presented graphically in Fig. 6.

One way to learn about and foretell how well a classification model will perform on a set of test data where the actual values are known is to look at a confusion matrix. In most cases, a confusion matrix will show both the real and predicted values of a classification matrix. Accuracy, sensitivity, specificity, and error rate are computed from the confusion matrix once it has been constructed for every algorithm that has been implemented. The confusion matrices for the algorithms in our proposed work are listed below in Fig. 8. The results show that the deep learning based stacking classifier achieves the highest accuracy. The addition of blockchain ensures secure and verifiable data handling. In Table 10, we have compared accuracy of our proposed work with

Table 7
Performance metrics of different classifiers.

Classifier	Accuracy	F1 score	Recall	Precision	Specificity	FPR	FDR	FNR	MCC	AuC
Logistic regression	0.75	0.7093	0.6931	0.7261	0.7946	0.2053	0.2738	0.3068	0.4906	0.8602
Naive bayes	0.705	0.6380	0.5909	0.6933	0.7946	0.2053	0.3066	0.4090	0.3953	0.8281
Random forest	0.87	0.8555	0.875	0.8369	0.8660	0.1339	0.1630	0.125	0.7380	0.9206
XGBoost	0.835	0.8176	0.8409	0.7956	0.8303	0.1696	0.2043	0.1590	0.6680	0.9052
KNN	0.805	0.7914	0.8409	0.7474	0.7767	0.2232	0.2525	0.1590	0.6132	0.8711
Decision tree	0.8	0.7802	0.8068	0.7553	0.7946	0.2053	0.2446	0.1931	0.5981	0.8007
SVM	0.75	0.7126	0.7045	0.7209	0.7857	0.2142	0.2790	0.2954	0.4915	0.8536
Stacking based	**0.905**	**0.9073**	**0.93**	**0.8857**	**0.88**	**0.12**	**0.1142**	**0.07**	**0.8110**	**0.9211**

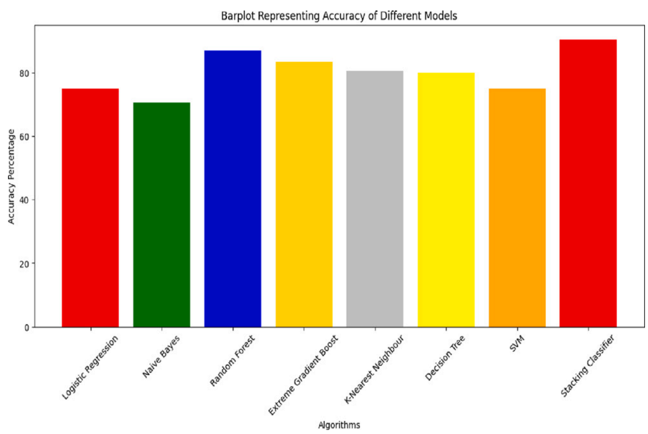


Fig. 5. Comparison of accuracy for various classifiers.

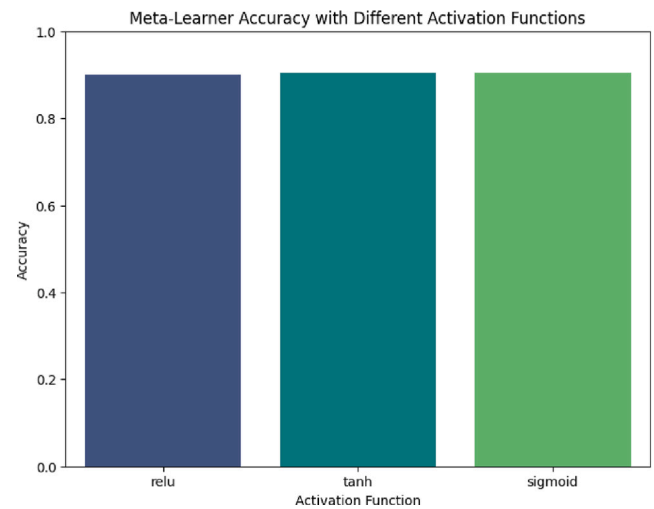


Fig. 6. Accuracy Comparison of Activation Functions.

Table 8
Stacking classifier permutations and accuracy results.

Index	Permutation	Accuracy
0	RF, SVM, LR, NB, XGBoost, KNN, DT	0.88
1	RF, SVM, LR, NB, XGBoost, DT, KNN	0.885
2	RF, SVM, LR, NB, KNN, XGBoost, DT	0.90
3	RF, SVM, LR, NB, KNN, DT, XGBoost	0.905
4	RF, SVM, LR, NB, DT, XGBoost, KNN	0.905
5	RF, SVM, LR, NB, DT, KNN, XGBoost	0.90
6	RF, SVM, LR, XGBoost, NB, KNN, DT	0.905

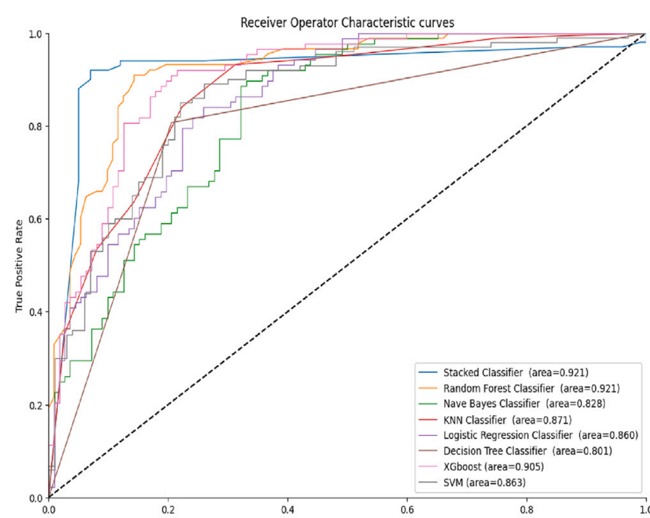


Fig. 7. Receiver operating characteristics for different classifier.

Table 9
Accuracy results for different activation functions.

Activation function	Accuracy
ReLU	0.9000
Tanh	0.9050
Sigmoid	0.9050

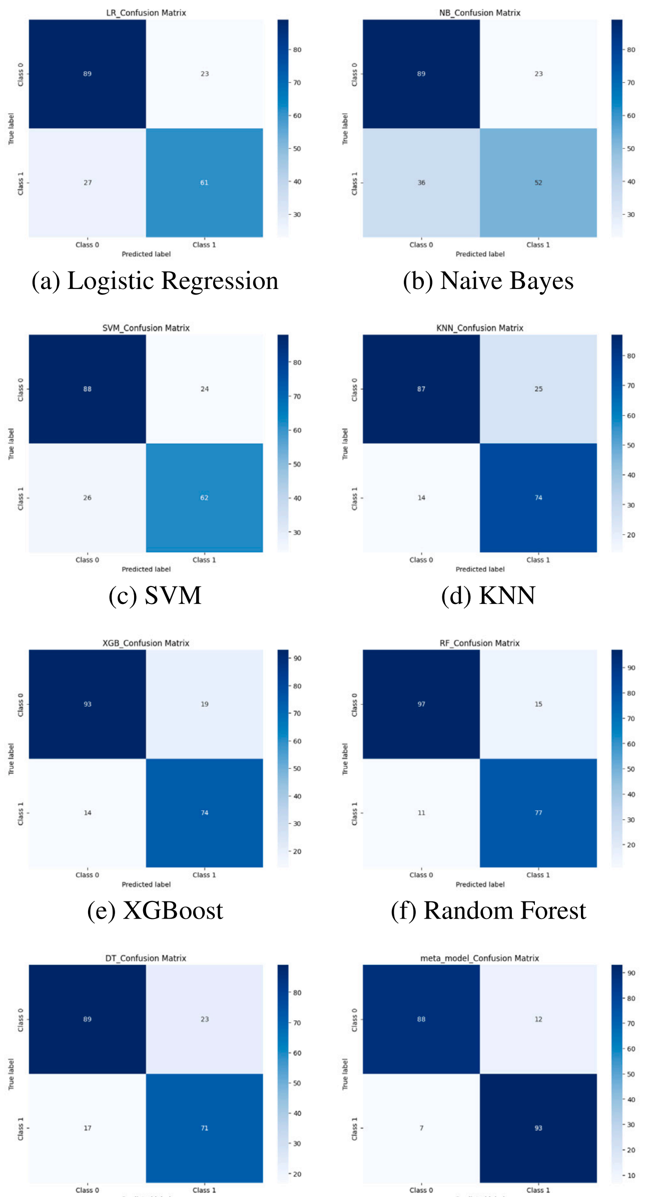
Table 10
Comparison of existing research studies.

S No.	Author(s)	Techniques used	Accuracy (%)
1	Khanam et al. [9]	NB	79.42
2	Alam et al. [7]	ANN	75.00
3	Sarwar et al. [14]	KNN	77.00
4	Shafi et al. [8]	SVM	74.28
5	Sainte et al. [13]	REPTree	74.48
6	Singh et al. [11]	NSGA-II	83.30
7	Ahmed et al. [6]	SVM	80.26
8	Kalagotla et al. [15]	Stacking	78.20
9	Zhou et al. [19]	Ensemble Model	88.00
10	**Proposed work**	**Stacked ML + Deep learning**	**90.50**

the accuracy of various researcher's work on same dataset. It can be seen clearly that our proposed techniques outperforms different existing works. Binary cross-entropy loss was applied because our classification task is binary (diabetic vs. non-diabetic), and it is the standard loss function for binary deep learning classifiers. The meta-learner was implemented using a deep learning model (a shallow neural network) while logistic regression, Naive Bayes, Random Forest, XGBoost, KNN, Decision Tree, and SVM are used as base learner. Logistic regression can also be used as meta-learners.

5.2. Blockchain implementation results

Table 11 presents the results of a throughput test conducted on a blockchain-based logging system under varying simulated transaction loads. At three levels of testing load — 10, 50, and 100 transactions — the system consistently maintained a throughput of approximately



(a) Logistic Regression (b) Naive Bayes (c) SVM (d) KNN (e) XGBoost (f) Random Forest (g) Decision Tree (h) Stacking Classifier

Fig. 8. Confusion matrices obtained from classifiers: (a) Logistic Regression, (b) Naive Bayes, (c) SVM, (d) KNN, (e) XGBoost, (f) Random Forest, (g) Decision Tree, and (h) Stacking Classifier.

Table 11
Blockchain throughput test results.

Testing load (Txns)	Simulation time (s)	TPS (Approx.)
10	0.50	19.97
50	2.51	19.95
100	5.02	19.93

20 transactions per second (TPS), with only minimal variation across trials (ranging from 19.93 to 19.97 TPS). This stability in performance suggests that the blockchain implementation is capable of handling increasing transaction volumes efficiently without significant degradation in processing speed, making it suitable for healthcare scenarios requiring reliable and predictable logging under moderate clinical loads.

Tamper evidence demo (hash logging)

The following Python script uses SHA-256 hashing to generate a tamper-evident log from patient record strings.

```python
import hashlib

def local_hash(data):
    return hashlib.sha256(data.encode()).
        ↪ hexdigest()

sample = "PID01:DiagnosisA:123456"
print("SHA256 Hash:", local_hash(sample))
```

Listing 1: SHA-256 based tamper-evident hash logging

Output:

Console Output

```
SHA256 Hash: affd65135a04003e12514d65d0afd5007d78
48970308079ad8c27aa2dc31c99a
```

Connection to Ganache or Infura and setup contract

```python
from web3 import Web3
import time

# Connect to Ethereum mainnet using
    ↪ Alchemy
w3 = Web3(Web3.HTTPProvider(
    "https://eth-mainnet.g.alchemy.com/v2/
        ↪ your-api-key"
))
print("Connected:", w3.is_connected())

# Define contract ABI and address (from
    ↪ Remix deployment)
abi = [
    # ABI goes here
]
```

Listing 2: Web3.py connection setup with Alchemy/Infura

Output:

Console Output

```
Connected: True
```

Blockchain transaction using web3.py

```python
# Create a transaction
nonce = w3.eth.get_transaction_count(
    ↪ account)
txn = contract.functions.logPrediction(
    "PatientID123", "DiabetesDetected",
        ↪ str(time.time())
).build_transaction({
    'from': account,
    'nonce': nonce,
    'gas': 200000,
    'gasPrice': w3.to_wei('0', 'gwei')
})

```

```
# Sign and send
signed_txn = w3.eth.account.sign_transaction(txn, private_key)
tx_hash = w3.eth.send_raw_transaction(signed_txn.raw_transaction)

print("Transaction successful!")
print("Tx Hash:", tx_hash.hex())
```

Listing 3: Smart contract invocation and transaction logging

Sample Output:

Console Output

```
Transaction successful! ✓
Tx Hash: 0xabcdef1234567890abcdef1234567890abcdef
1234567890abcdef1234567890
```

5.3. Clinical utility & economic feasibility

The cost–benefit analysis of the proposed StaBloCare system underscores multiple advantages for healthcare providers. Deploying the prediction model in the cloud guarantees scalable and safe access to diagnostic instruments. Implementing smart contracts on the blockchain offers immutable audit trails, hence improving data integrity and regulatory compliance. Timely prediction of diabetes aids in averting complications, thereby diminishing long-term treatment expenses. Moreover, automating the creation of diagnostic reports enhances productivity by conserving staff time and optimizing clinical operations. The system provides an efficient and economical approach for enhancing healthcare delivery.

6. Conclusion & future scope

In this study, we used a stacked deep learning classifier and multiple machine learning classifiers to predict diabetes on the PIDD Dataset. For usage in our classifiers — Gaussian NB, DT, KNN, SVM, XGBoost, and RF — it was then divided into training and testing sets and standardized. We have created a secure and impenetrable system that uses blockchain technology to safeguard patient data throughout the prediction process in order to address the risks of data breaches, unauthorized access, and manipulation that come with traditional centralized systems. Accuracy, F1 score, precision, and recall were among the measures used to assess each classifier's performance. With an accuracy score of 0.905 and an AuC score of 0.9211, the results show that the best performance is achieved when machine learning classifiers are stacked with deep learning classifiers. All things considered, this experiment demonstrates how machine learning methods may be applied to accurately forecast diabetes, which can aid medical professionals in the early detection and prevention of the condition. ML-based diabetes diagnosis has a bright future ahead of it, with enormous potential to enhance healthcare outcomes. Diabetes can be detected earlier and more precisely because of machine learning algorithms' ability to analyze vast volumes of data and spot patterns that humans might miss. A sizable dataset with thousands of records with no missing values is needed to create a prediction model with a higher accuracy in diabetes prediction. The integration of additional techniques into the employed model will be the main focus of our upcoming work. A large dataset with few or no missing attribute values will allow the testing model to record more insights and improve forecast accuracy. All things considered, the potential for diabetes detection through machine learning is enormous, and further study and development in this field could transform diabetes diagnosis and treatment, improving patient outcomes and making better use of available healthcare resources. There is a practical feasibility of EHR integration pathways with common hospital systems (Epic, Cerner). This can be done by integrating pathway using FHIR (Fast Healthcare Interoperability Resources). This integration can export prediction results in FHIR-compatible JSON format. These can be integrated with hospital EHRs via secure REST APIs, making our model interoperable with existing workflows. Future work involving longitudinal EHR data is needed for true risk prediction modeling. We would also like to explore FedML and other privacy-preserving alternatives in the future.

7. Limitations

Despite showing encouraging findings with a 90.5% accuracy rate utilizing a stacked deep learning model, the suggested StaBloCare system has a few drawbacks. It is entirely dependent on the PIMA Indian Diabetes Dataset, which limits generalizability due to its small sample size and lack of demographic variety. Blockchain integration also adds processing overhead and possible latency problems, even though it improves data security. The prediction consistency is further impacted by the stacking method's sensitivity to the base learners' order. Furthermore, there is still a lack of research on practical deployment issues such sensor dependability, network scalability, and energy expenses.

CRediT authorship contribution statement

Monu Bhagat: Writing – original draft, Visualization, Validation, Conceptualization. **Ujjwal Maulik:** Writing – review & editing, Supervision.

Code availability

The Dockerized implementation of *StaBloCare* is available at: https://github.com/monubhagat11iitkgp/StaBloCare.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors gratefully acknowledge the handling editor and the anonymous reviewers for their insightful and constructive feedback, which significantly enhanced the quality and clarity of this manuscript. Dr. Monu Bhagat extends his sincere thanks to Prof. Prabir Kumar Biswas, Head of the Department of AI and Director, AI4ICPS Innovation Hub, IIT Kharagpur, for his continuous support, guidance, and provision of research resources.

Abbreviations and acronyms

PIDD	Pima Indian Diabetes Dataset
ML	Machine Learning
SMOTE	Synthetic Minority Oversampling Technique
HIPAA	Health Insurance Portability and Accountability Act
BMI	Body Mass Index
NB	Naïve Bayes
RF	Random Forest
DT	Decision Tree
SVM	Support Vector Machine
LR	Logistic Regression
KNN	K-Nearest Neighbor
K-Means	K-Means Clustering
NN	Neural Networks
RNN	Recurrent Neural Network
GB	Gradient Boost
MLP	Multilevel Perceptron
AB	AdaBoost
XGB	XGBoost
LDA	Linear Discriminant Analysis

ETC	Extra Trees Classifier
NSGA-II	Non-dominated Sorting Genetic Algorithm II
ROC	Receiver Operating Curve
AuC	Area Under ROC Curve
EOA	Externally Owned Accounts

Data availability

Data will be made available on request.

References

[1] M. Javaid, A. Haleem, R.P. Singh, R. Suman, S. Rab, Significance of machine learning in healthcare: Features, pillars and applications, Int. J. Intell. Netw. 3 (2022) 58–73.
[2] S.S. Bhat, V. Selvam, G.A. Ansari, M.D. Ansari, M.H. Rahman, Prevalence and early prediction of diabetes using machine learning in North Kashmir: a case study of district Bandipora, Comput. Intell. Neurosci. (2022).
[3] Global Burden of Disease Collaborative Network, Global Burden of Disease Study, Institute for Health Metrics and Evaluation (IHME), 2020.
[4] National Health Mission, MoHFW, Government of India, National health mission, 2020, https://nhm.gov.in.
[5] M.J. Muckley, et al., Results of the 2020 fastMRI challenge for machine learning MR image reconstruction, IEEE Trans. Med. Imaging 40 (9) (2021) 2306–2317, http://dx.doi.org/10.1109/TMI.2021.3075856.
[6] N. Ahmed, R. Ahammed, M.M. Islam, M.A. Uddin, A. Akhter, M.A. Talukder, B.K. Paul, Machine learning based diabetes prediction and development of smart web application, Int. J. Cogn. Comput. Eng. 2 (2021) 229–241.
[7] T.M. Alam, M.A. Iqbal, Y. Ali, A. Wahab, S. Ijaz, T.I. Baig, A. Hussain, et al., A model for early prediction of diabetes, Inform. Med. Unlocked 16 (2019) 100204.
[8] S. Shafi, G.A. Ansari, Early prediction of diabetes disease and classification of algorithms using machine learning approach, in: International Conference on Smart Data Intelligence, ICSMDI, 2021.
[9] J.J. Khanam, S.Y. Foo, A comparison of machine learning algorithms for diabetes prediction, ICT Express 7 (4) (2021) 432–439.
[10] A. Mujumdar, V. Vaidehi, Diabetes prediction using machine learning algorithms, Procedia Comput. Sci. 165 (2019) 292–299.
[11] N. Singh, P. Singh, Stacking-based multi-objective evolutionary ensemble framework for prediction of diabetes mellitus, Biocybern. Biomed. Eng. 40 (1) (2020) 1–22.
[12] M.A.R. Refat, M.A. Amin, C. Kaushal, M.N. Yeasmin, M.K. Islam, A comparative analysis of early stage diabetes prediction using machine learning and deep learning approach, in: 6th International Conference on Signal Processing, Computing and Control, ISPCC, IEEE, 2021, pp. 654–659.
[13] S. Larabi-Marie-Sainte, L. Aburahmah, R. Almohaini, T. Saba, Current techniques for diabetes prediction: review and case study, Appl. Sci. 9 (21) (2019) 4604.
[14] M.A. Sarwar, N. Kamal, W. Hamid, M.A. Shah, Prediction of diabetes using machine learning algorithms in healthcare, in: 24th International Conference on Automation and Computing, ICAC, IEEE, 2018, pp. 1–6.
[15] S.K. Kalagotla, S.V. Gangashetty, K. Giridhar, A novel stacking technique for prediction of diabetes, Comput. Biol. Med. 135 (2021) 104554, http://dx.doi.org/10.1016/j.compbiomed.2021.104554.
[16] K. Abnoosian, R. Farnoosh, M.H. Behzadi, Prediction of diabetes disease using an ensemble of machine learning multi-classifier models, BMC Bioinformatics 24 (1) (2023) 337.
[17] K. Abnoosian, R. Farnoosh, M.H. Behzadi, A pipeline-based framework for early prediction of diabetes, J. Heal. Biomed. Inform. 10 (2) (2023) 125–140.
[18] R. Farnoosh, K. Abnoosian, A robust innovative pipeline-based machine learning framework for predicting COVID-19 in Mexican patients, Int. J. Syst. Assur. Eng. Manag. 15 (7) (2024) 3466–3484.
[19] H. Zhou, S. Rahman, M. Angelova, C.R. Bruce, C. Karmakar, A robust and generalized framework in diabetes classification across heterogeneous environments, Comput. Biol. Med. 186 (2025) 109720.
[20] M. Chen, T. Malook, A.U. Rehman, Y. Muhammad, M.D. Alshehri, A. Akbar, M. Bilal, M.A. Khan, Blockchain-enabled healthcare system for detection of diabetes, J. Inf. Secur. Appl. 58 (2021) 102771.
[21] A. Karmakar, P. Ghosh, P.S. Banerjee, D. De, ChainSure: Agent free insurance system using blockchain for healthcare 4.0, Intell. Syst. Appl. 17 (2023) 200177.
[22] A. Gupta, M. Bhagat, V. Jain, Blockchain-enabled healthcare monitoring system for early Monkeypox detection, J. Supercomput. 79 (14) (2023) 15675–15699.
[23] D. Sisodia, D.S. Sisodia, Prediction of diabetes using classification algorithms, Procedia Comput. Sci. 132 (2018) 1578–1585.
[24] N. Sneha, T. Gangil, Analysis of diabetes mellitus for early prediction using optimal feature selection, J. Big Data 6 (1) (2019) 1–19.
[25] S.M. Ganie, M.B. Malik, Comparative analysis of various supervised machine learning algorithms for the early prediction of type-II diabetes mellitus, Int. J. Med. Eng. Informatics 14 (6) (2022) 473–483.
[26] S. Kumari, D. Kumar, M. Mittal, An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier, Int. J. Cogn. Comput. Eng. 2 (2021) 40–46.
[27] R.D. Joshi, C.K. Dhakal, Predicting type 2 diabetes using logistic regression and machine learning approaches, Int. J. Environ. Res. Public Heal. 18 (14) (2021) 7346.
[28] H. Lu, S. Uddin, F. Hajati, M.A. Moni, M. Khushi, A patient network-based machine learning model for disease prediction: The case of type 2 diabetes mellitus, Appl. Intell. 52 (3) (2022) 2411–2422.
[29] Q. Zou, K. Qu, Y. Luo, D. Yin, Y. Ju, H. Tang, Predicting diabetes mellitus with machine learning techniques, Front. Genet. 9 (2018) 515.
[30] M. Maniruzzaman, M.J. Rahman, B. Ahammed, M.M. Abedin, Classification and prediction of diabetes disease using machine learning paradigm, Heal. Inf. Sci. Syst. 8 (2020) 1–14.
[31] Y. Zheng, R. Lu, S. Zhang, J. Shao, H. Zhu, Achieving practical and privacy-preserving kNN query over encrypted data, IEEE Trans. Dependable Secur. Comput. (2024).
[32] Y.E. Zein, M. Lemay, K. Huguenin, Privatree: Collaborative privacy-preserving training of decision trees on biomedical data, IEEE/ACM Trans. Comput. Biol. Bioinform. 21 (1) (2023) 1–13.
[33] S. Alex, D.K. Jagalchandran, D.P. Pattathil, Privacy-preserving and energy-saving random forest-based disease detection framework for green Internet of Things in mobile healthcare networks, IEEE Trans. Dependable Secur. Comput. 21 (4) (2023) 4180–4192.
[34] X. Ding, H. Zhang, C. Ma, X. Zhang, K. Zhong, User identification across multiple social networks based on naive Bayes model, IEEE Trans. Neural Netw. Learn. Syst. 35 (3) (2022) 4274–4285.
[35] D. Berrar, Bayes' theorem and naive Bayes classifier, in: Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics, 2018, pp. 403–412.
[36] M.S. Aldeen, C. Zhao, Z. Chen, L. Fang, Z. Liu, Privacy-preserving collaborative learning for genome analysis via secure XGBoost, IEEE Trans. Dependable Secur. Comput. (2024).
[37] C.N. Li, Y. Li, Y.H. Shao, Large-scale structured output classification via multiple structured support vector machine by splitting, IEEE Trans. Emerg. Top. Comput. Intell. 8 (2) (2024) 2112–2124.
[38] C. Thuppari, S. Jannu, D.R. Edla, A.K. Mishra, D. Gautam, A. Vidyarthi, D.G. Aray, A novel k-coverage and m-connectivity based fire prediction model for consumer IoT using fish swarm optimization and logistic regression model, IEEE Trans. Consum. Electron. (2024).
[39] R. Singhal, R. Kashef, A weighted stacking ensemble model with sampling for fake reviews detection, IEEE Trans. Comput. Soc. Syst. 11 (2) (2023) 2578–2594.
[40] M. Rehman, I.T. Javed, K.N. Qureshi, T. Margaria, G. Jeon, A cyber secure medical management system by using blockchain, IEEE Trans. Comput. Soc. Syst. 10 (4) (2022) 2123–2136.
[41] C.L. Blake, C.J. Merz, UCI repository of machine learning databases, 1998, http://archive.ics.uci.edu/ml/index.php.

Dr. Monu Bhagat is currently serving as an Assistant Professor in the Department of Computer Science and Engineering at Birla Institute of Technology, Mesra, Ranchi, Jharkhand, India. Prior to joining BIT Mesra, he was a Chanakya Postdoctoral Fellow in the Department of Artificial Intelligence and AI4ICPS Innovation Hub at the Indian Institute of Technology, Kharagpur. He received his Ph.D. in Computer Science and Engineering from the National Institute of Technology Jamshedpur, following an M.Tech in Computer Science and Engineering from Kalyani Government Engineering College and a B.Tech in Computer Science and Engineering from the West Bengal University of Technology. Dr. Bhagat has over seven years of academic and research experience, having served at institutions such as Techno Main Salt Lake (Kolkata), Manipal University Jaipur, NIT Jamshedpur, and IIT Kharagpur. He has published more than 25 research articles in reputed journals, international conference proceedings, book chapters, and abstracts, and holds four national and international patents. He has supervised three M.Tech theses and over 30 B.Tech projects. His research interests include Machine Learning, Image Processing, Blockchain, Explainable AI, Internet of Things, and Generative AI.

Prof.(Dr.) Ujjwal Maullik is a Professor in the Dept. of Comp. Sc. and Engg., Jadavpur University since 2004. Dr. Maulik has worked in many universities and research laboratories around the world as visiting Professor/ Scientist. Currently he is serving as the Visiting Professor in the department of Electrical Engineering, Standford University, Stanford, California, United States. He is the recipient of Alexander von Humboldt Fellowship during 2010, 2011 and 2012 and Senior Associate of ICTP, Italy during 2012–2018 and Fulbright Fellowship in 2024–2025. He is the Fellow of Indian National Academy of Engineers (INAE), India, National Academy of Science India (NASI), International Association for Pattern Recognition (IAPR), USA, The Institute of Electrical and Electronics Engineers (IEEE), USA and Asia-Pacific Artificial Intelligence Association (AAIA), Hongkong. He is also the Distinguish Member of the ACM. He is a Distinguish Speaker of IEEE as well as ACM. His research interest include Machine Learning, Pattern Analysis, Data Science, Bioinformatics, Multi-objective Optimization, Social Networking, IoT and Autonomous Car. In these areas he has published ten books, more than four hundred papers.