A Blockchain and ML-Based Framework for Fast and Cost-Effective Health Insurance Industry Operations

Anubhav Elhence, Adit Goyal, Vinay Chamola, *Senior Member, IEEE*, and Biplab Sikdar, *Senior Member, IEEE*

***Abstract*— Health insurance is crucial for each person, bearing in mind the increasing medical costs. COVID-19 has been an eye-opener as to how important it is to have health insurance. Medical emergencies can have a severe emotional and financial impact. Thus, a health insurance policy can help mitigate financial risks in unpredictable circumstances. However, the current insurance system is very expensive, as thousands of people pay the premiums, and very few take the claims. Furthermore, the claim settlement process is excruciatingly long and tiresome. In this article, we focus on establishing a rapid and cost-effective framework for the health insurance market, based on machine learning and blockchain technology. By developing a smart contract, blockchain may eliminate any third-party organizations and make the complete process safer, easier, and more efficient. The contract pays the claim based on the claimant's documentation. We optimized the premiums using a regression model based on the net amount claimed during the current policy tenure and various other criteria. For anticipating risk, a random forest classifier is used, which aids in the risk-rated premium rebate computation for policyholders for their next term of insurance.**

***Index Terms*— Blockchain, Ethereum, insurance, machine learning, random forest, regression, smart contract.**

I. Introduction

THE cost of healthcare services has been rising exponentially since the introduction of COVID-19 and other rising cases of infections. COVID-19 has caused millions of deaths and forced a large number of patients to be admitted in the hospital for treatment [1]. One cannot predict medical exigencies, and it can disrupt one's entire financial planning [2]. In the situation of a pandemic with a high infection rate, no successful vaccine, and medical treatment inflation, the importance of a good health insurance plan cannot be more evident. It provides a good backup in times of unavoidable and expensive medical services.

An insurance plan is an optimal financial strategy, as it shares the risk where one pays a small premium consistently and can claim a large amount of assured money during an emergency. On the other hand, the minimal and steady premium has seen a price increase under the existing insurance system. The insurance industry spends a significant amount of money on sales and administrative costs, which include commissions to brokers, commercials, brochures, and other general running costs, which are subsequently recovered through insurance premiums. The traditional insurance method has numerous flaws. For example, the insurer must complete a large amount of paperwork and produce verification of the loss's worth. The broker adds to the delays and costs. Because the insurer's knowledge sources are diverse and inconsistent, one must seek further information. Claim analysis is subjective and manual [3]. At the time of a health emergency, the last thing a policyholder wants is a delay in the insurance payment procedure. Another problem prevalent in the traditional insurance system is when people keep on paying high premiums without making claims for a long time. These premiums only go into the profit of the insurance companies. Thus, the large number of middlemen in the insurance sector has made the whole process expensive and inefficient.

One of the earliest and most sought-after cryptocurrencies is Bitcoin [4]. Its technological advantage is that it allows secure transactions without an integrated management system. It has received much consideration and advancement in the field of academic research [5], [6], [7], [8], [9]. A blockchain's architecture is that of a block consisting of multiple transactions which join in a chain-like fashion to a preceding block [10], [11]. A blockchain time-stamp ledger is used to distribute and save data in a shared way [12]. It is tamper-resistant [13]. Bitcoin and Litecoin are examples of blockchains used to store data as payment records, and Ethereum is used to store data as contracts or private data [14], [15].

Smart contracts have long been a notion, but with the start of blockchain technology, particularly the Ethereum blockchain [16], and the growth of the programming language Solidity [17], they have started to move from just an idea to reality. Smart contracts can also be written in Python using the SmartPy library by Smart Chain Arena. Blockchain uses

Manuscript received 6 June 2022; revised 28 August 2022 and 18 October 2022; accepted 22 October 2022. Date of publication 10 November 2022; date of current version 2 August 2023. The work of Vinay Chamola and Biplab Sikdar was supported by the Association of Southeast Asian Nations (ASEAN)—India Collaborative Research and Development Scheme through the ASEAN-India Science & Technology Development Fund (AISTDF) (Sponsored) under Grant CRD/2020/000369. This work was supported by the Ministry of Education, Singapore under Grant R-263-000-D63-114. *(Corresponding author: Vinay Chamola.)*

Anubhav Elhence and Vinay Chamola are with the Department of Electrical and Electronics Engineering and APPCAIR, Birla Institute of Technology and Science, Pilani (BITS Pilani), Pilani 333031, India (e-mail: elhenceanubhav@gmail.com; vinay.chamola@pilani.bits-pilani.ac.in).

Adit Goyal is with the Department of Computer Science and IT, Jaypee Institute of Information Technology, Noida 201304, India (e-mail: aditgoyal@hotmail.com).

Biplab Sikdar is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077 (e-mail: bsikdar@nus.edu.sg).

Digital Object Identifier 10.1109/TCSS.2022.3219256

cloud technology to reduce expenditure and provides efficient and more secure transactions by automating all the back-end operations [3]. Blockchain was welcomed by industry leaders and the World Economic Forum in 2016. It was at that time when the improvements that blockchain can bring to insurance processes were first identified [18].

Design Goals: In traditional systems, money transfers are slow, as they require many settlement processes. Because of the multiple main elements it pertains to, this study intends to use blockchain technology to find answers to our problem through the use of smart contracts, automating and expediting the operations of user registration, insurance policy issuing, and claim settlement. It employs a distributed, decentralized ledger capable of obviating the need for any third-party authentication. Because data are immutable and visible, fraud detection becomes easier to detect. The document's integrity is preserved via a transparent ledger of changes, which instills confidence in the asset. Furthermore, different people have different medical needs based on their personal backgrounds and situation. Keeping this in mind, it wouldn't be optimal to have a generalized insurance policy with a fixed premium for everyone. On top of this, the policyholders indirectly pay for all the middlemen staff in the entire system. To reduce the overall costs of insurance and make the insurance premium justified for each individual user, we proposed a blockchain-based network that uses machine learning techniques of ridge regression and random forest. It helps to determine the user's risk factor and subsequently calculate the premium individually for that policyholder.

The following are the significant contributions of our work.

1) A peer-to-peer network of hospitals and policyholders is proposed, which is bound by a smart contract. The smart contracts would also save each user's claim histories for the following term's premium computation.
2) We have divided the logic of the smart contract into six phases: client registration phase, client query phase, policy initialization phase, policy issuing phase, policy claiming phase, and policy renewal phase. Corresponding to each phase of the smart contract, a pseudocode explaining the functions of the smart contract is provided.
3) The premium is predicted using ridge regression. The admitting hospital sends the claims straight to the smart contract, which approves them if all the conditions are met.
4) Before renewing their policy for the next term, the customer's risk is forecast using a random forest classifier. If the contract's value is sufficient for expected claims, all policyholders may be eligible for a rebate depending on their expected risk. To this end, a risk-rated premium rebate model for the following term's premium computation has been proposed.
5) When compared to other methods like XGBoost, experimental results suggest that the proposed framework is more accurate, quick, and cost-effective.

The rest of this article is organized as follows. Section II presents the related work. Section III presents the system model, while Section IV describes the proposed framework. Results to evaluate the performance of the proposed system are presented in Section V, and Section VI concludes this paper.

II. Related Work

The insurance industry has been investigating blockchain technology through substantial investments in its research and uses in various sub-operations [19], [20], [21]. Over 40 organizations joined the B3i-oriented blockchain insurance consortium in 2016 [22]. The Cat XoL program that the B3i consortium is developing uses Corda Distributed Ledger Software. There was a hackathon conducted in Zurich where several insurers, brokers, and re-insurers participated, and this application was tested [23]. During the 4-h negotiation process, 1033 messages were generated, which led to the signing of 41 contracts.

In 2017, Etherisc, a German insurance business, made an open-source Decentralized Insurance Framework focusing on decentralized insurance applications [24]. Etherisc mainly focused on reducing inefficiencies like high fees for processing documents and large settlement time for claims by using blockchain. Beenest and WeTrust are two American insurance businesses that established a blockchain-based homeowners' insurance system in 2017 [25]. Guardtime, a software company from Estonia, and the logistics giant, Maersk, created a blockchain-based Insurance platform Insurewave [3]. Globally, Maersk integrates and retains networks of several data sources relevant to maritime insurance. Raikwar and Mazumdar [27] proposed using blockchain to automate insurance operations and eliminate the need for a third-party organization. The peer-to-peer blockchain paradigm has been applied in a variety of industries, including traffic congestion management [27], biometrics, and parking system security measures [28], [29], and also in insurance sectors such as cyber [30], travel [31], and automobile [32].

Predictive modeling has helped mathematicians and insurance analyzers in the insurance industry to enhance their task efficiencies. Premium computation for auto insurance companies [33], and risk classification for insurance claims [34] are done using predictive modeling. Regression has also been used in the financial and economic domains [35]. The authors of [36] used regression and classification methods for real-estate price prediction. Various studies have developed insurance pricing models and risk premium functions, which have contributed to improving the system's accuracy [37], [38]. Furthermore, Baser and Apaydin [39] proposed using hybrid fuzzy least square analysis for calculating insurance claim reserves.

The need for a cost-effective insurance system has increased with the inflation of medical expenses, and this article is motivated by the lack of research in this area. The need to eliminate extra costs that go to all the third-party bearers, which make the whole process slow, less secure, and more costly, encouraged us to use blockchain for health insurance, which has shown outstanding performance in other insurance sectors. Furthermore, to learn the complex relationships in various parameters of insurance processes, we use ridge regression because of its promising results in this field. Ridge regression can handle a large volume of uncertainty, inaccuracy, and lack

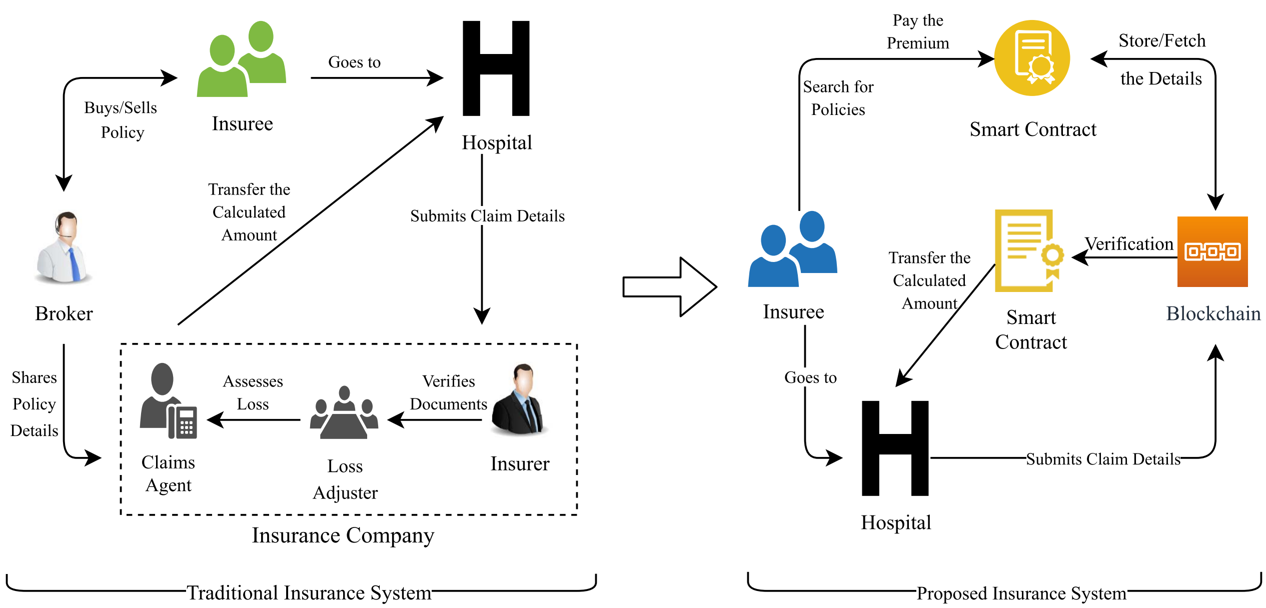

Fig. 1. Traditional insurance system versus proposed insurance system.

of data [40]. Due to the optimal results on multiclass classification, ensemble approaches like the random forest, XGBoost, or decision trees are employed in numerous frameworks for risk classification [41], [42], [43]. For insurance data analysis, the developers of [44] employed an ensemble random forest classifier, and the suggested framework outperformed existing algorithms like support vector machine (SVM).

III. System Model

We propose a peer-to-peer blockchain network replacing the insurance industry from the middle through a smart contract. The smart contract, along with the network, would be monitored and maintained by the insurance company intending to remove the middle-man structure and provide a seamless experience to their policyholders. A comparison between the classical insurance system process and our proposed blockchain-based insurance process is shown in Fig. 1. The implications of the proposed system model are as follows.

1) There are two types of peers that can enter into the network.
    a) A person intending to purchase a policy to insure himself against health emergencies can register into the blockchain network by providing basic details such as name, age, and gender. Insurance industries are not able to completely utilize the benefits of machine learning due to complex underlying rule sets and non-KYC (know your customer) environments. Hence, we consider the primary factors associated with any individual that tend to affect the pattern of claims made against their policy.
    b) A hospital intending to provide medical services to policyholders can join the insurance network. Hospitals must provide their registration certificate, a list of doctors, and their specializations and degrees. The hospital can then become an in-network hospital.

    Each account is identifiable with a unique set of account addresses, a public key, and a private key, after it is formed (of type *a* or *b*).
2) After registering, a person can view the predicted premium amount based on factors such as their age, gender, BMI, smoking history, region, and the number of children. These parameters are passed to a machine learning model which uses regression to predict the premium value.
3) If the calculated premium is acceptable to the user, a smart contract is formed with the peer (policyholder), and the policy will be issued that covers the hospitalization expenses.
4) After the policy issuance, the policyholder pays the calculated premium of that term into the smart contract from its dashboard via Ethereum.
5) When policyholders need to avail medical services, they can view the in-network hospitals and directly go to them and provide the hospitals with their public key. The hospital will send the claim process for approval to the smart contract by filling in the policyholder's unique public key, details of the treatment, additional charges (if any), such as intensive care unit (ICU) charges and outpatient department (OPD) charges, pharmaceutical expenses, name of treating doctor and the hospital's unique public key.
6) The claim is processed through the smart contract, which will check the authenticity of the treating doctor from the back-end of the hospital registration, and all the expenses will be verified and calculated.
7) After the claim is approved, the amount will be directly transferred from the smart contract to the hospital.
8) The smart contract will store the number of claims made by all the policyholders. After the end of the policy term, it will re-evaluate the risk of the insured party by passing its information through a trained model. The result will give the risk on a scale of 1–8. The risk is

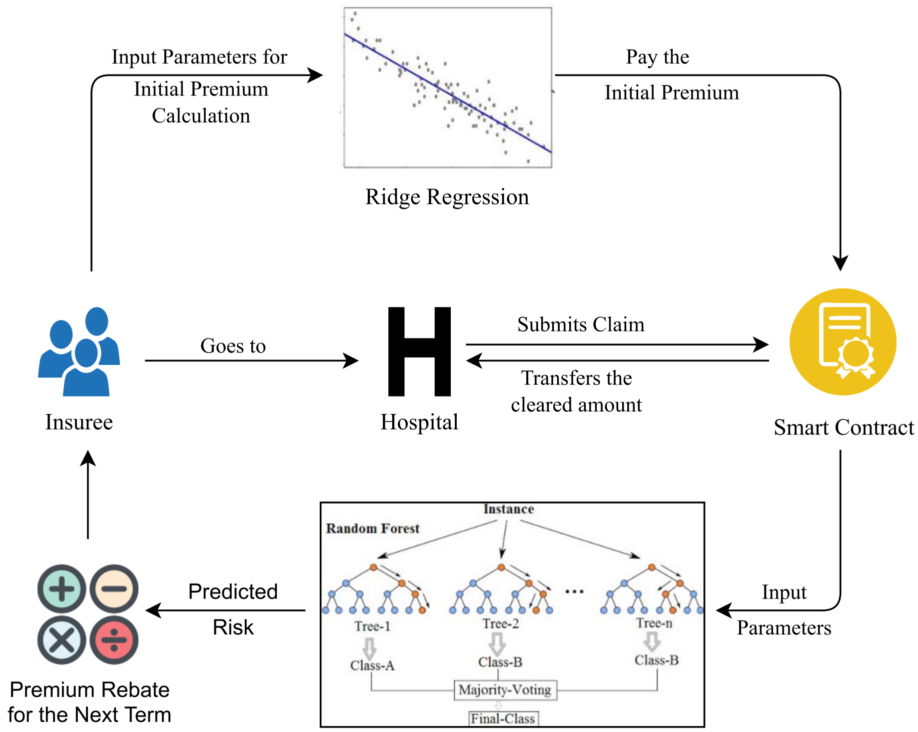

Fig. 2. Proposed model's flowchart.

calculated based on medical history, age, gender, and various other factors. The total amount of money in the smart contract (at the current stage of calculating: premiums collected—claim amount paid), the number of claims filed by the insured party, and the risk model result will be supplied as parameters to a risk-rated premium rebate mathematical model. This model will determine the revised premium after factoring in the rebate.

9) The new premium calculated after the risk-rated premium rebate will be reflected in the insured party's dashboard from where they can pay the premium for their next term of insurance.

The next section discusses the proposed models and the mathematical formulations involved in the premium calculation and premium rebates.

IV. Proposed Model

Consider the case of a person P, who fills out the necessary information to join the blockchain network. The health insurance policy's premium is predicted using a regression model. When P is taken to a hospital, H, the hospital completes all the necessary paperwork for processing the claim and automated settlement of the claim. When a policy term ends, a random forest classifier model is used to determine an individual's risk, which is then given to the risk-rated premium rebate mathematical model that is used to calculate the updated policy premium. The proposed model's flowchart is presented in Fig. 2.

A. Prerequisite Knowledge Required

1) Ridge Regression: Any data that exhibit multicollinearity can be analyzed using the model-tuning technique known as ridge regression. Ridge regression is similar to linear regression, but the variance of the estimates is reduced by using a regularized version of the model through a cost function. This method performs L2 regularization. The response of this regression model is obtained from a probability (Gaussian or normal) distribution, unlike linear regression, where the output is estimated as a single value.

2) Random Forest: Decision trees are responsive to the particular data that are used to train them. If the training data are updated, the outcomes of the decision tree can be quite different. Decision trees are computationally costly, have a risk of overfitting, and also tend to find local optima, because they cannot go back after splitting. Random forests are used to fix these limitations. Many decision trees are trained together to produce a single output using the ensemble (multiple models combined) model technique known as random forest. This merging of decision trees is termed as bagging. Due to its advantage of combining the output of all decision trees, it results in high accuracy.

B. Ridge Regression for Premium Prediction

The output of similar approaches employing the entire collection of features as model inputs was tested using linear regression as the baseline model. L_2 regularization adds a penalizing factor to the squared error cost function to assist the method in converging to linearly separable data and mitigating overfitting. The cost function that must be minimized in relation to a parameter, θ, is as follows:

$$J(\theta) = \|y - X\theta\|_2^2 + \alpha\|\theta\|_2^2 \tag{1}$$

where α is a hyperparameter and X is a design matrix. Because the baseline model has a lot of variance, ridge regression would be the ideal choice.

TABLE I
DATASET'S INPUT AND OUTPUT PARAMETERS FOR PREMIUM PREDICTION

Input Fields	**Data Type**	**Description**
Age	Integer	Age of Customer.
Sex	String	Customer's Gender (female/male).
BMI	Numeric	Ratio of height to weight (Body mass index).
Children	Integer	No. of children.
Smoker	String	Whether or not the customer has smoking history (no/yes).
Region	String	Residential region of the customer.
Output Fields	**Data Type**	**Description**
Premium Amount	Numeric	Insurance premium for the policy.

The US Health Insurance dataset used here contains 1338 rows of data without any undefined or missing entries [45]. We verified each feature of the dataset for initial prepossessing to convert the Boolean parameters into binary and categorical features into integer-encoded vectors. After preprocessing, the feature vector is made up of six pieces. Furthermore, the characteristics were standardized to ensure that all of the data fell within the same range. Table I lists the input and output parameters in detail.

Data normalization is crucial in scaling up the raw data to a standard format for efficient prediction. Normalization is done on each attribute by considering the minimum and maximum values in that column using

$$A_{\text{new}} = X_{\min} + \frac{A_t - A_{\min}}{A_{\max} - A_{\min}}(X_{\max} - X_{\min}). \tag{2}$$

In (2), the value of A_t is normalized to A_{new}. $A_{\min}$ and $A_{\max}$ are the minimum and maximum values of the attribute column, respectively, and $X_{\max}$ and $X_{\min}$ are the maximum and minimum values of the normalization range (usually $[0-1]$), respectively.

C. Risk Calculation Using Random Forest Classification

After the policy is issued, the person becomes a policyholder and submits his/her premium amount into the smart contract. After a term, the policyholder's risk is assessed based on various factors, including insurance history, family history, job history, and medical history. However, with health insurance, in particular, the probability of a claim depends on several factors of a policyholder, and there is a very nonlinear relationship between the characteristics of the insured party and the sum of the claims they made. To give a rebate in the coming term's premium, a risk factor of each individual covered is determined as indicated in Table II. A random forest classifier predicts this risk factor.

In this model, we use various parameters for input, as listed in Table III. These input parameters are fed into the random forest classifier model. Random forest is an ensemble learning method widely used for regression and classification purposes, as shown in Algorithm 1 [46]. The hyperparameters to be tuned for the algorithm are n_estimators and max _features. n_estimators indicates the number of trees built before aggregating the predictions and max _features indicates the maximum number of features to take into account while splitting a node. The space of potential splitting features is divided into a random subset of a specific size, and the best feature is deterministically chosen from that subset. To classify a data point, the random forest classifier combines the classification result from each of the trees using an aggregate function. It can perform large-scale machine learning tasks effectively. This algorithm was chosen due to its superior performance, easy-to-use python interface, as well as optimal time and memory usage. Owing to the nonlinear relationship between the price and the feature vector, the regression tree was chosen as the appropriate model for this problem.

TABLE II
RISK CORRESPONDING TO PREDICTED OUTPUT

Risk Factor	**Predicted Value**
No risk	8
Very low	7
Low	6
Medium-low	5
Medium	4
Medium-high	3
High	2
Very high	1

In this study, we use the Prudential Life Insurance Dataset [47] containing 59 381 rows of data. The data are divided into two sets: training and testing, comprising of 53 442 and 5939 rows, respectively. We removed the attributes that were only utilized to identify the recipients during preprocessing. To express categorical features in a numeric vector, they were factorized. In addition, a new function that displays the number of medical keywords has been included. After preprocessing step, the feature vector includes ten features.

D. Premium Calculation for the Following Term Using Risk-Rated Premium Rebate Model

To optimize the insurance process, a premium rebate rated on risk is a feasible option that can be provided to all policyholders. To calculate our refund based on premium, we apply the concept of weighted (risk-rated) premium computation [48]. The present term's premium is $\mathcal{P}_x$, and the next term's premium, $E[x]$, must be determined. The medical policy covers risk x, and the claims we receive from x are distributed as a random variable with a cumulative distribution function $F(x)$. The properties of risk-rated premium calculation that we use here are as follows.

1) *Nonnegative Loading:* It states that $\mathcal{P}_x \geq E[x]$, i.e., the risk-rated premium amount will not be less than the estimated amount. If the risk x is constant, then the loading is zero.
2) *Additivity:* If $x1$ and $x2$ are independent risks, then the combined risk-rated premium $\mathcal{P}_{x1+x2}$ should be equal to

TABLE III
DATASET'S INPUT AND OUTPUT FEATURES FOR RISK PREDICTION

Input Features	**Interpretation**
BMI	Body Mass Index of the customer.
Age	Policyholder's age.
Weight	Policyholder's weight.
Height	Policyholder's height.
Insured Information	Normalised features' set related to existing insurance details.
Employment Information	Normalised variables' set related to the work history.
Medical History	Normalised parameters' set linked to the policyholder's medical history.
Family History	Normalised parameters' set linked to the policyholder's family history.
Insurance History	Normalised parameters' set related to the previous insurance details, if any.
Medical Keywords	A set of parameters indicating a particular disease's presence/absence.
Output Features	**Interpretation**
Risk Factor	Target variable that forecasts the risk within the range $[1, 8]$.

Algorithm 1 Random Forest Classifier

Data: D training data, $n\_estimators$: number of trees, $max\_features$: maximum number of features to take into account while splitting

```
// To generate trees:
```
$i \leftarrow 0$;
while $i < n\_estimators$ **do**
  Randomly sample $\mathcal{D}$ with replacement to produce $\mathcal{D}_i$;
  Create root node, $\mathcal{N}_i$, containing $\mathcal{D}_i$;
  Call `BuildTree(`$\mathcal{N}_i$`);`
  $i+=1$;
end
BuildTree($\mathcal{N}$):
if *$\mathcal{N}$ contains instances of only one class* **then**
  return;
else
  Randomly select $x\%$ of the possible splitting features ($\leq max\_features$) in $\mathcal{N}$;
  Select the feature $\mathcal{F}$ with the highest information gain to split on;
  Create f child nodes of $\mathcal{N}$, $\mathcal{N}_1, \ldots, \mathcal{N}_f$, where $\mathcal{F}$ has f possible values ($\mathcal{F}_1, \ldots, \mathcal{F}_f$);
  $i \leftarrow 1$;
  while $i \leq f$ **do**
    Set the contents of $\mathcal{N}_i$ to $\mathcal{D}_i$, where $\mathcal{D}_i$ is all instances in $\mathcal{N}$ that match $\mathcal{F}_i$;
    Call `BuildTree(`$\mathcal{N}_i$`);`
    $i+=1$;
  end
end

$\mathcal{P}_{x1} + \mathcal{P}_{x2}$. In this situation, the overall premium charges do not change by combining or dividing the risk.

3) *Consistency:* This property states that we should have $\mathcal{P}_Y = \mathcal{P}_X + b$, if $F(Y) = F(X) + b$. Therefore, if the distribution of Y is X's distribution moved by b units, then the risk-rated premium Y will be the risk-rated premium X increased by b.

Let $\mathcal{C}$ be the entire amount in the smart contract at the moment. For the calculation of expected claims in the next term of the policy, we compute the risk of each policyholder making the claims. In the range $[1, 8]$, a risk factor of 1 corresponds to claims for the full amount covered (let the sum insured be $\mathcal{I}$), while a risk factor of 8 indicates no claim. The cumulative amount of money spent on the expected claims would be $\mathcal{T}$

$$\mathcal{T} = \sum_{i=1}^{n} \frac{8 - r_i}{7} \mathcal{I} \tag{3}$$

where r_i is individual i's associated risk, and n indicates how many policyholders are registered in the contract. The excess revenue (ER) that is collected is then calculated

$$\mathrm{ER} = \mathcal{C} + A - \mathcal{T} \tag{4}$$

where A is the entire amount received from premiums, considering that no rebate is offered to those customers whose policy term has to be re-insured in the present situation. The risk-rated premium is estimated for each user, and the excess revenue collected from policyholders is used to pay a refund. The risk-adjusted premium principle can be calculated as follows:

$$\mathcal{P}_x = \int_0^{\infty} [1 - F(x)]^{1/r} dx. \tag{5}$$

The premium in (5) is based on Esscher's principle [49]. The Esscher principle translates weights into the distribution of the risk x, giving weight to the tail (right) probabilities to increase their influence. Similarly, the risk-adjusted premium is based on this principle. The distribution function G of a positive random variable x^* is defined as

$$1 - G(x^*) = [1 - F(x)]^{1/r}. \tag{6}$$

Since

$$E[x^*] = \int_0^{\infty} [1 - G(x)] dx \tag{7}$$

then, $\mathcal{P}_x = E[x^*]$. We consider here that x is an exponentially distributed function with a mean of $1/\lambda$. We then have

$$1 - F(x) = e^{-\lambda x} \tag{8}$$
$$1 - G(x^*) = e^{-\lambda x/r}. \tag{9}$$

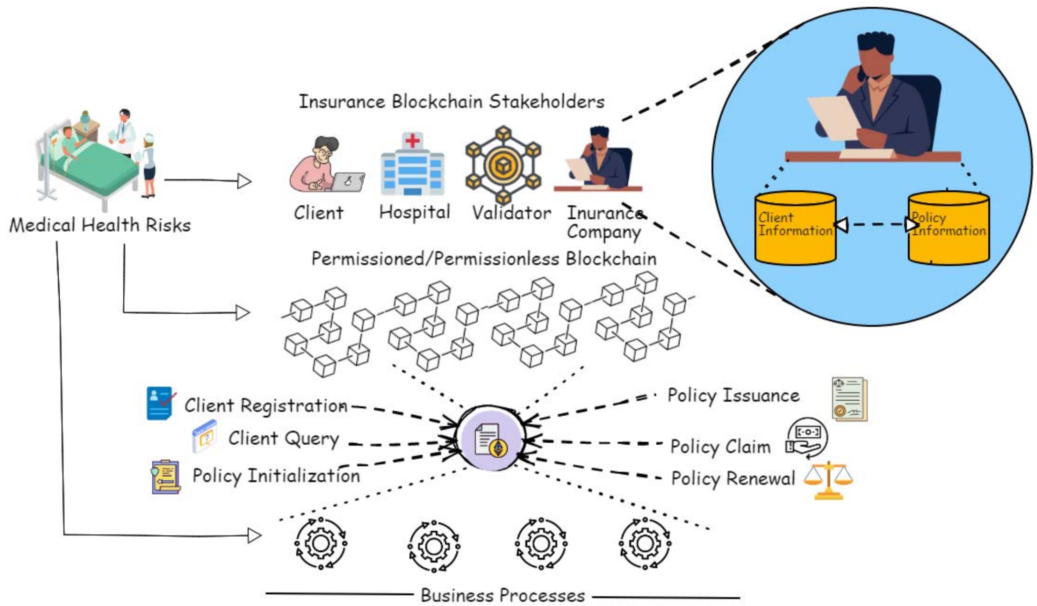

Fig. 3. Insurance blockchain network architecture.

Thus, x^* has an exponential distribution with a mean of r/λ. Since r from the aforementioned risk model is in decreasing order: 1 being the highest risk and 8 being the lowest risk, and in (9), it is in ascending order, we now replace r with $9 - r$. Thus

$$\mathcal{P}_x = \frac{9 - r}{\lambda}. \tag{10}$$

The total risk associated with all the individuals is

$$\mathcal{R} = \sum_{i=1}^{n} r_i. \tag{11}$$

Hence, the risk-rated premium rebate is

$$\mathcal{P}_r = \mathcal{P}_x - \frac{r}{R}\text{ER}. \tag{12}$$

This will appear on the policyholder's dashboard, allowing them to keep their policy for the next term based on their risk level.

E. Smart Contract Outline and Specifications

Using the Ethereum smart contract platform, the insurance details will be contractualized. The insurance information, along with policy data, covered asset information, coverage terms, coverage period, previous claim history, and client information, goes into the smart contract. The stakeholders in the smart contract are the clients, the insurance company, hospitals, and validators (Fig. 3). The client is the main character around whom the smart contract algorithm revolves. The client can register himself onto the blockchain network in the client registration phase, can buy an insurance policy in the policy issuance phase, can apply for a refund in the policy claim phase, and can also apply for renewal of insurance in the policy renewal phase. The insurance company releases various policies in the policy initialization phase. The client can choose the policy which he wants to buy after seeing the predicted premium value generated from the machine learning model.

TABLE IV

DEFINITIONS OF NOTATIONS USED IN THE ALGORITHMS

Notation	Definition
EA	Ethereum Address
CEA	Client Ethereum Address
HEA	Hospital Ethereum Address
SCEA	Smart Contract Ethereum Address
CI	Client information like name, age, BMI, sex, etc.
UCI	Updated Client Information
PI	Policy Information like policy id, policy name, etc.
IPA	Initial Premium Amount
UPA	Updated Premium Amount
MTD	Medical treatment Details
MLMRC	Machine Learning Model for Risk Calculation
MLMPP	Machine Learning Model for Premium Prediction
RRPRM	Risk Rated Premium Rebate Model

Fig. 4 shows the communication flow among the various characters involved in the smart contract. A list of notations used in the algorithms is mentioned in Table IV.

1) Client Registration: The insurance company calls a smart contract to register the client into the insurance blockchain network. The smart contract has a client object structure, $\text{Struct}_{\text{client}}$, which has attributes like client identification (CI) information, ETH address of the client (CEA), BMI, age, and other parameters as mentioned in Table I. The smart contract creates a client object $\text{Obj}_{\text{client}}$ from the client object structure $\text{Struct}_{\text{client}}$. The algorithm for client registration process is discussed in Algorithm 2.

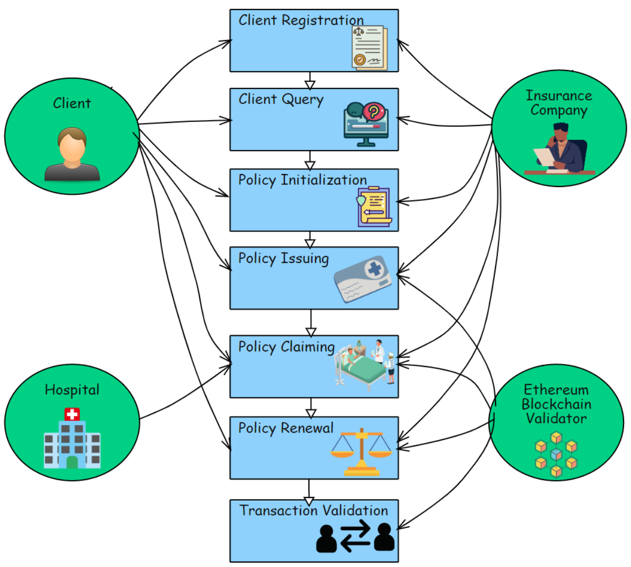

Fig. 4. Communication flow among the stakeholders in the smart contract.

Algorithm 2 Client Registration

function REGISTERCLIENT(CEA, CI)
 $Struct_{client} \leftarrow$ (EA, name, age, BMI, sex, children, smoker, region);
 Check and verify the details of client (CEA, CI);
 $Obj_{client} \leftarrow$ f(CEA, CI);
 Store (CEA, Obj_{client}) in database;
 Emit an event to insurance company about successful registration of the client;
end function

2) Client Query: Once the client is registered into the insurance blockchain network, the insurance company can retrieve all the client attributes. To retrieve the client information, the insurance company has to query the blockchain using the CEA. The algorithm for client query process is discussed in Algorithm 3.

Algorithm 3 Client Query

function QUERYCLIENT(CEA)
 Search (CEA) in the database;
 if *exists* **then**
  retrieve Obj_{client} and return;
 else
  return error;
 end
end function

3) Policy Initialization: The smart contract has a policy object structure $\text{Struct}_{\text{Policy}}$ having policy information (PI) such as policy id, policy name, policy terms and conditions, ML model for risk calculation (MLMRC) and premium prediction (MLMPP), and other metadata related to the policy. When the insurance company releases a new policy, a policy object $\text{Obj}_{\text{Policy}}$ is created. While initializing a policy, a policy-client object structure is also created, which stores the policy purchase details and the insurance utilization details by the client. The attributes of this structure include CEA, policy id, the amount claimed, claim submission date, claim acceptance indicator, etc. The algorithm for policy initialization process is discussed in Algorithm 4.

Algorithm 4 Policy Initialization

function INITIALIZEPOLICY($PI, MLMRC, MLMPP$)
 $Struct_{Policy} \leftarrow$ (Policy ID, Policy Name, Policy terms and conditions, ML models);
 $Obj_{policy} \leftarrow$ Insurance company releases new policy $g(PI, MLMRC, MLMPP)$;
 $Struct_{policy-client} \leftarrow$ (EA, Policy ID, premium_amt, claimed_amt, claim_date);
 Store Obj_{policy} in database;
 Emit an event to insurance company about successful release of policy Obj_{policy};
end function

4) Policy Issuing: A client can choose the policy from the available policies in the smart contract. Corresponding to each policy, the client can see the initial premium amount (IPA) generated by the ML model, which he has to pay to purchase the policy. After choosing the policy, the client submits the premium into the smart contract. A corresponding policy-client object is created once the transaction passes all the verification and checks and is validated by the validators in the Ethereum network. The policy-client object $\text{Obj}_{\text{policy-client}}$ is queryable using the unique policy issuance id, and the object gets linked to the client object and the policy object. The algorithm for policy issuing process is discussed in Algorithm 5.

Algorithm 5 Policy Issuing

function ISSUEPOLICY(Obj_{policy}, Obj_{client})
 Check if $Obj_{policy-client}$ already exists;
 Calculate IPA $\leftarrow$ MLmodel (Obj_{policy}, Obj_{client});
 Check if Obj_{client} satisfies the terms and conditions of Obj_{policy};
 Client sends IPA amount of tokens from CEA to Smart Contract Ethereum Address (SCEA);
 if *transaction == verified* **then**
  Create $Obj_{policy-client} \leftarrow$ new $Struct_{policy-client}$ (EA, Policy ID, premium_amt, claimed_amt, claim_date);
 else
  return Error;
 end
 Store $Obj_{policy-client}$ in database;
 Emit an event to insurance company about successful issuance of new policy;
end function

5) Policy Claiming: When the policyholder goes to a hospital registered in the network and avails medical services, the hospital files for the claim from the smart contract by submitting the CEA, Hospital Ethereum Address (HEA), medical treatment details (MTD), medical expenses, and the final amount to be claimed. The smart contract processes the claim by verifying all the conditions and smart checks and calculations. Once approved, the amount is directly transferred from the smart contract to HEA. The algorithm for the policy claiming process is discussed in Algorithm 6.

Algorithm 6 Policy Claim

function CLAIMPOLICY(CEA, HEA, MTD)
  Check if $Obj_{policy-client}$ already exists;
  $Struct_{claim}$ ← (CEA, HEA, medical expenses, MTD, verification Documents, requested claim amount);
  Obj_{claim} ← new $Struct_{claim}$(CEA, HEA, MTD);
  Check if Obj_{claim} is valid or not;
  if Obj_{claim} == *accepted* **then**
    Calculate the client_reimburse_amount based on $Obj_{policy-client}$;
    if *claimed_amount + client_reimburse_amount <= Policy_reimburse* **then**
      Send client_reimburse_amount amount of tokens from SCEA to HEA;
    else
      Send Policy_reimburse - claimed_amount amount of tokens from SCEA to HEA;
    end
  else
    return Error;
  end
  Emit an event to insurance company, client and hospital about successful claim of insurance policy;
end function

6) Policy Renewal: The policy-client object stores the information regarding the utilization of the policy such as the amount claimed and claim submission. After the end of the term of any policy, the smart contract will re-evaluate the risk of the client by passing the data through a trained model. The model will give the risk on a scale of 1–8 based on the medical history, age, gender, BMI, and other factors. Using the risk-rated premium rebate model (RRPRM), the smart contract will determine the revised premium after factoring in the rebate. The client can see the updated premium on his dashboard and can opt to renew the policy. The algorithm for policy renewal process is discussed in Algorithm 7.

V. Numerical Analysis and Results

A. Simulation Settings

On the blockchain network, a smart contract is used as the first node that cannot be changed, and the blockchain is utilized to create the peer-to-peer network. Smart contracts are created in the Python programming language and deployed on the Ethereum network using the SmartPy module. The machine learning models are also implemented in Python using various libraries such as TensorFlow, pandas, scikit-learn, and NumPy. To increase the ridge regression model's performance, we employed polynomial features of degree = 7 and a value of $\alpha = 3$ to get the optimum results. The training and testing sets for ridge regression are split in a 3:1 ratio.

The training and testing datasets are split 9:1 at random in the risk calculation algorithm. The number of trees in the forest

Algorithm 7 Policy Renewal

function RENEWPOLICY(Obj_{policy}, Obj_{client}, Updated Client Information)
  Check if $Obj_{policy-client}$ needs to be renewed or not;
  Calculate Updated Premium Amount (UPA) ← RRPRM(Obj_{policy}, Obj_{client}, UCI);
  Check if Obj_{client} satisfies the updated terms and conditions of Obj_{policy};
  Client sends UPA amount of tokens from CEA to SCEA;
  if *transaction* == *verified* **then**
    Update $Obj'_{policy-client}$ ← $Obj_{policy-client}$ (CEA, Policy ID, premium_amt, claimed_amt, claim_date);
  else
    return Error;
  end
  Update $Obj'_{policy-client}$ in database;
  Emit an event to insurance company about successful renewal of policy;
end function

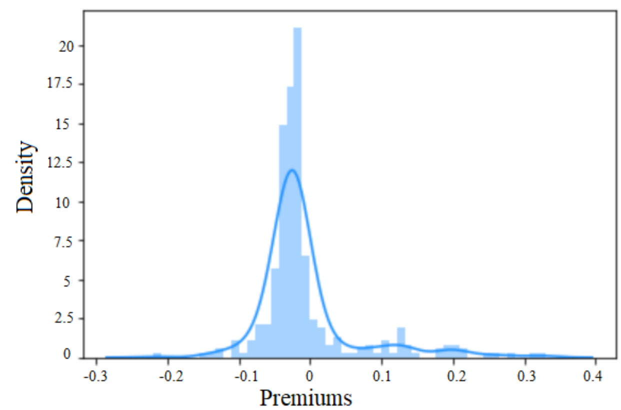

Fig. 5. Kernel density estimation plots help visualize the difference between actual and predicted values on the normalized test set for the ridge regression model.

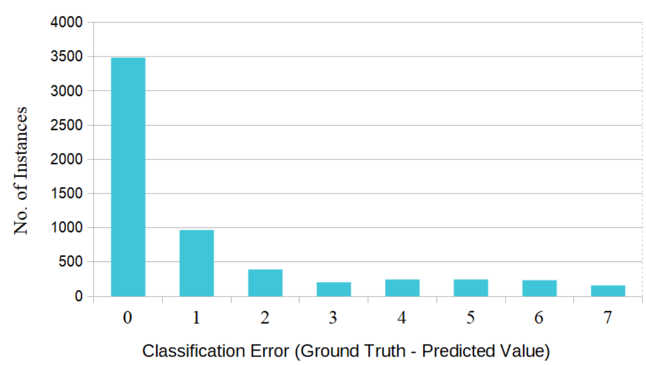

Fig. 6. For the random forest classifier, histogram density plots are quantifying the difference between actual and predicted values on the testing set.

(n_estimators) is set to 50, the minimum number of data points at the leaf node (min_samples_leaf) is set to 8, the number of features to take into account for splitting (max_features) is set to 75, the minimum number of samples required to split an internal node (min_samples_split) is set to 40, the maximum

TABLE V
PERFORMANCE COMPARISON OF DIFFERENT MODELS USED FOR RISK CALCULATION

Classification Model	Root Mean Square Error • Training data • Testing data	Quadratic Weighted Kappa • Training data • Testing data	Mean Absolute Error • Training data • Testing data	Accuracy Score • Training data • Testing data
Random Forest	2.03 2.26	0.61 0.5	0.92 1.19	0.71 0.59
XGBoost	2.5 2.63	0.65 0.61	2.4 2.18	0.28 0.27

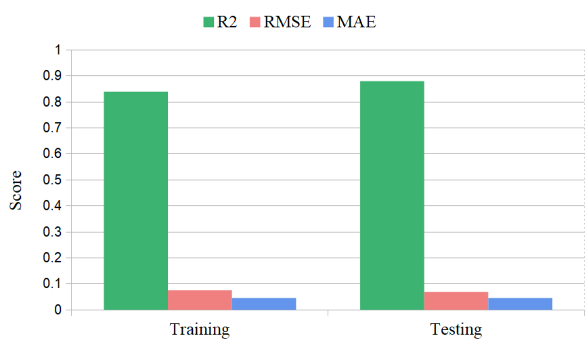

Fig. 7. Ridge regression model's performance metrics.

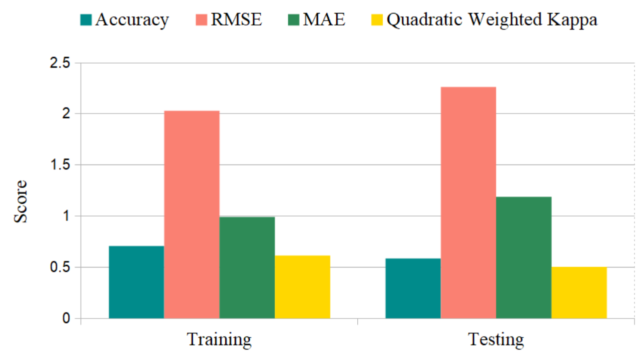

Fig. 8. Random forest classifier's performance metrics.

depth of the tree (max_depth) is set to 50, and the bootstrap is set to False. The default values for all other parameters are maintained. The entire code-base for the proposed framework is available at: https://github.com/Adit31/A-Blockchain-and-ML-based-Framework-for-Health-Insurance.

B. Performance Evaluation

Mean absolute error (MAE) is the average of the absolute errors in the predicted values [50]. RMSE values indicate how close the data points are to the best-fit line. A low value of RMSE means that the predicted values deviate less from the regression line, or in other words, the predicted values are close to the actual values. Quadratic weighted Kappa values range from -1 to 1, where a value of 1 indicates that all predicted values match with the expected values, and -1 indicates that none of the predicted values were right and are farthest from the actual values. A score greater than 0.5 is considered fairly good [51]. To test the performance of the model, F1 scores are used along with precision and recall. Each class corresponds to the risk level associated with it, i.e., class 1 means a risk value of 1 (very high risk).

Fig. 5 depicts the kernel density estimation plot for the ridge regression model, whereas Fig. 6 depicts the histogram density plot for the random forest classification. The MAE, root-mean-square error (RMSE), and R^2 scores were obtained for premium prediction, as shown in Fig. 7. Ridge regression with interaction terms had the greatest R^2 and the lowest RMSE value of all the regression models examined. To analyze the performance of our model in the risk classification task, we calculate the RMSE, quadratic weighted kappa score, MAE, and accuracy, as shown in Fig. 8 and Table V (all values were computed on results in the range [1, 8]). Moreover, F1 scores have also been calculated for each class in the training and testing sets and are depicted in Fig. 9.

A huge number of classes are responsible for the low accuracy. The predicted classes are within a few percentage points of the actual number (Fig. 6), as seen by the low MAE. The Quadratic Weighted Kappa score was calculated to measure the agreement between the predicted and the actual classes. Quadratic weighted value of 0.61 on the training set and 0.5 on the testing set show that the predicted risk values are in close range to the expected values. All the metrics collectively depict that the model proposed can classify the users in their respective risk classes using their personal information such as age, BMI, and sex, as parameters.

In the comparison of cost, in the traditional model, the premiums tend to be high due to extra revenue generated by the insurance companies and various third parties involved, which also generate their compensations. In the proposed model, we have eliminated all third parties and reduced the extra revenues. In the traditional method, each claim goes through many levels of agents that approve and forward the claim and then the claim settling agent that transfers the amount. However, in the proposed model, a smart contract saves all this processing time and is very fast. As soon as the hospital fills in the details, it processes the claim from the back-end and settles it. On comparing the rebate, the existing method provides no claim bonus, but the expenditure remains the same. The reflection is only during a complete claim, whereas in the proposed model, the individual receives rebates on premium deposits with lower risk.

With the integration of IoMT devices with blockchain network, we can utilize the devices to either make a GRPC call directly from the network or we can use a routing protocol like PEERP to fetch electronic health records of the client [52]. This data can be used as an input to MLMRC, MLMPP, and RRPRM. Irrespective of the machine learning model used,

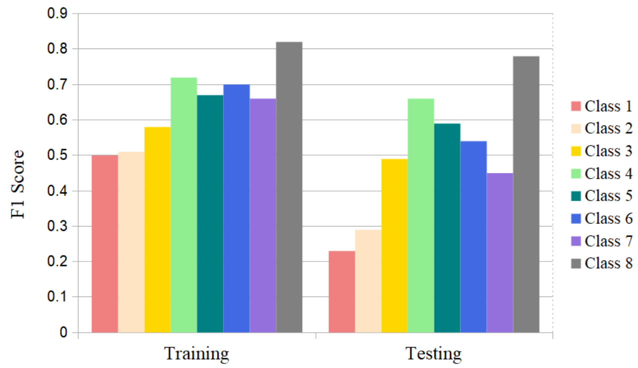

Fig. 9. F1-score of the random forest classifier.

whether it is random forest classifier or multiclass classifier ML model like XGBoost, the integration of IoMT devices with the blockchain network will be done in similar manner using any secure data aggregation scheme [53]. Since the patient health records are directly involved, data aggregation scheme has to be privacy preserving and maintain the integrity and authenticity of the data. Also, if the communication overheads can be reduced making the scheme lightweight and efficient, it will contribute to improving the bandwidth and latency of the blockchain network [54].

VI. Conclusion

The suggested blockchain-based health insurance model solves the two key flaws in the present system that are expensive and slow. The results demonstrate that our proposed methodology is accurate, cost-effective, and quick. With fewer intermediaries and smart contracts, the entire claim processing process may be completed quickly, and risk-rated premium rebates can encourage consumers with fewer claims to stay with their insurance. It also puts a stop to the extra income generated by the current insurance system by allocating it properly via risk-based rebates. The results reveal that the random forest classifier outperformed other multiclass classification machine learning algorithms like XGBoost.

References

[1] B. Aslam, A. R. Javed, C. Chakraborty, J. Nebhen, S. Raqib, and M. Rizwan, "Blockchain and ANFIS empowered IoMT application for privacy preserved contact tracing in COVID-19 pandemic," *Pers. Ubiquitous Comput.*, pp. 1–17, Jul. 2021, doi: 10.1007/s00779-021-01596-3.

[2] L. Ennaceur, S. B. Othman, C. Chakraborty, F. A. Almalki, and H. Sakli, "Emerging technologies to combat the COVID-19 pandemic," in *Digital Health Transformation with Blockchain and Artificial Intelligence*. Boca Raton, FL, USA: CRC Press, 2022, pp. 235–251.

[3] V. Aleksieva, H. Valchanov, and A. Huliyan, "Application of smart contracts based on ethereum blockchain for the purpose of insurance services," in *Proc. Int. Conf. Biomed. Innov. Appl. (BIA)*, Nov. 2019, pp. 1–4.

[4] S. Nakamoto. (2018). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[5] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "SoK: Research perspectives and challenges for bitcoin and cryptocurrencies," in *Proc. IEEE Symp. Secur. Privacy*, May 2015, pp. 104–121.

[6] K. M. Abiodun et al., "Blockchain and in healthcare systems: Prospects, issues, and challenges," in *Digital Health Transformation with Blockchain and Artificial Intelligence*. Boca Raton, FL, USA: CRC Press, 2022, pp. 1–22.

[7] K. Yu, L. Tan, X. Shang, J. Huang, G. Srivastava, and P. Chatterjee, "Efficient and privacy-preserving medical research support platform against COVID-19: A blockchain-based approach," *IEEE Consum. Electron. Mag.*, vol. 10, no. 2, pp. 111–120, Mar. 2021.

[8] R. Chaudhary, A. Jindal, G. S. Aujla, N. Kumar, A. K. Das, and N. Saxena, "LSCSH: Lattice-based secure cryptosystem for smart healthcare in smart cities environment," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 24–32, Apr. 2018.

[9] G. Bansal, V. Hasija, V. Chamola, N. Kumar, and M. Guizani, "Smart stock exchange market: A secure predictive decentralized model," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[10] H. Watanabe, S. Fujimura, A. Nakadaira, Y. Miyazaki, A. Akutsu, and J. Kishigami, "Blockchain contract: Securing a blockchain applied to smart contracts," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2016, pp. 467–468.

[11] V. Hassija, G. Bansal, V. Chamola, V. Saxena, and B. Sikdar, "BlockCom: A blockchain based commerce model for smart communities using auction mechanism," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2019, pp. 1–6.

[12] A. Yazdinejad, G. Srivastava, R. M. Parizi, A. Dehghantanha, K.-K.-R. Choo, and M. Aledhari, "Decentralized authentication of distributed patients in hospital networks using blockchain," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 8, pp. 2146–2156, Aug. 2020.

[13] V. Hassija, V. Saxena, and V. Chamola, "A mobile data offloading framework based on a combination of blockchain and virtual voting," *Softw., Pract. Exper.*, vol. 51, no. 12, pp. 2428–2445, Dec. 2021.

[14] L. Zhou, L. Wang, and Y. Sun, "MIStore: A blockchain-based medical insurance storage system," *J. Med. Syst.*, vol. 42, no. 8, p. 149, Aug. 2018.

[15] M. Singh et al., "CovaDel: A blockchain-enabled secure and QoS-aware drone delivery framework for COVID-like pandemics," *Computing*, vol. 104, pp. 1589–1613, Mar. 2022.

[16] *Ethereum Homestead Documentation*. Accessed: Dec. 20, 2022. [Online]. Available: https://ethdocs.org/en/latest/

[17] *Introduction to Smart Contracts*. Accessed: Dec. 20, 2022. [Online]. Available: https://solidity.readthedocs.io/en/develop/introduction-to-smart-contracts.html

[18] G. Bruno, "The future of financial infrastructure. An ambitious look at how blockchain can reshape financial services," in *World Economic Forum*. World Economic Forum in Collaboration With Deloitte, 2016, p. 130.

[19] A. Adavoudi Jolfaei, S. F. Aghili, and D. Singelee, "A survey on blockchain-based IoMT systems: Towards scalability," *IEEE Access*, vol. 9, pp. 148948–148975, 2021.

[20] V. Hassija, G. Bansal, V. Chamola, N. Kumar, and M. Guizani, "Secure lending: Blockchain and prospect theory-based decentralized credit scoring model," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2566–2575, Oct./Dec. 2020.

[21] K. N. Mishra and C. Chakraborty, "A novel approach toward enhancing the quality of life in smart cities using clouds and IoT-based technologies," in *Digital Twin Technologies and Smart Cities*. Switzerland: Springer, 2020, pp. 19–35.

[22] B3i. *Digital Transformation for Reinsurance is Here*. Accessed: Aug. 7, 2021. [Online]. Available: https://b3i.tech/home.html

[23] Bea. *Hackathon–Zurich*. Accessed: Aug. 7, 2021. [Online]. Available: https://beamediagroup.com/projects/hackathon-event-zurich/

[24] Etherisc. (2017). *Etherisc-Decentralized Insurance Protocol to Collectively Build Insurance Products*. Accessed: Aug. 9, 2021. [Online]. Available: https://etherisc.com/

[25] G. Newswire. (2018). *Bee Token Partners With Wetrust to Strengthen Insurance Offering in New Home-Sharing Network Powered by Blockchain*. Accessed: Aug. 9, 2021. [Online]. Available: https://www.globenewswire.com/en/news-release/2018/01/26/1313129/0/en/Bee-Token-Partners-With-WeTrust-To-Strengthen-Insurance-Offering-In-New-Home-Sharing-Network-Powered-by-Blockchain.html

[26] M. Raikwar, S. Mazumdar, S. Ruj, S. Sen Gupta, A. Chattopadhyay, and K.-Y. Lam, "A blockchain framework for insurance processes," in *Proc. 9th IFIP Int. Conf. New Technol., Mobility Secur. (NTMS)*, Feb. 2018, pp. 1–4.

[27] V. Hassija, V. Gupta, S. Garg, and V. Chamola, "Traffic jam probability estimation based on blockchain and deep neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 3919–3928, Jul. 2021.

[28] A. Waheed and P. Venkata Krishna, "Comparing biometric and blockchain security mechanisms in smart parking system," in *Proc. Int. Conf. Inventive Comput. Technol. (ICICT)*, Feb. 2020, pp. 634–638.

[29] V. Hassija, V. Saxena, V. Chamola, and F. R. Yu, "A parking slot allocation framework based on virtual voting and adaptive pricing algorithm," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5945–5957, Jun. 2020.

[30] S. Feng, Z. Xiong, D. Niyato, P. Wang, S. S. Wang, and Y. Zhang, "Cyber risk management with risk aware cyber-insurance in blockchain networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.

[31] L. Jia-Lan, W. Xiao-Yu, Y. Wan-Jun, W. Zi-Chen, Z. Huai-Lin, and C. Nai-Meng, "Research and design of travel insurance system based on blockchain," in *Proc. Int. Conf. Intell. Informat. Biomed. Sci. (ICIIBMS)*, Nov. 2019, pp. 121–124.

[32] M. Demir, O. Turetken, and A. Ferworn, "Blockchain based transparent vehicle insurance management," in *Proc. 6th Int. Conf. Softw. Defined Syst. (SDS)*, Jun. 2019, pp. 213–220.

[33] M. David, "Auto insurance premium calculation using generalized linear models," *Proc. Econ. Finance*, vol. 20, pp. 147–156, Jan. 2015.

[34] P. Shi, X. Feng, and A. Ivantsova, "Dependent frequency–severity modeling of insurance claims," *Insurance: Math. Econ.*, vol. 64, pp. 417–428, Sep. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167668715001183

[35] S. Salcedo-Sanz, L. Cuadra, A. Portilla-Figueras, S. Jiménez-Fernández, and E. Alexandre, "A review of computational intelligence algorithms in insurance applications," in *Statistical and Soft Computing Approaches in Insurance Problems*. New York, NY, USA: Nova Science Publishers, 2012, pp. 1–50.

[36] H. Yu and J. Wu, "Real estate price prediction with regression and classification," Mach. Learn., Project Final Rep. CS 299, 2016.

[37] N. Krecar, F. E. Benth, and A. F. Gubina, "Towards definition of the risk premium function," *IEEE Trans. Power Syst.*, vol. 35, no. 2, pp. 1085–1098, Mar. 2020.

[38] H. Mao and K. Ostaszewski, "Pricing insurance contracts and determining optimal capital of insurers," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manage.*, Dec. 2010, pp. 1–5.

[39] F. Baser and A. Apaydin, "Calculating insurance claim reserves with hybrid fuzzy least squares regression analysis," *Gazi Univ. J. Sci.*, vol. 23, no. 2, pp. 163–170, 2010.

[40] J. B. A. Ghosh and J. A. Kumar, "End-to-end encryption and prediction of medical insurance cost," in *Proc. 6th Int. Conf. Trends Electron. Informat. (ICOEI)*, Apr. 2022, pp. 846–850.

[41] W. F. Mustika, H. Murfi, and Y. Widyaningsih, "Analysis accuracy of XGBoost model for multiclass classification—A case study of applicant level risk prediction for life insurance," in *Proc. 5th Int. Conf. Sci. Inf. Technol. (ICSITech)*, Oct. 2019, pp. 71–77.

[42] K. C. Dewi, H. Murfi, and S. Abdullah, "Analysis accuracy of random forest model for big data—A case study of claim severity prediction in car insurance," in *Proc. 5th Int. Conf. Sci. Inf. Technol. (ICSITech)*, Oct. 2019, pp. 60–65.

[43] N. K. Frempong, N. Nicholas, and M. Boateng, "Decision tree as a predictive modeling tool for auto insurance claims," *Int. J. Statist. Appl.*, vol. 7, no. 2, pp. 117–120, 2017.

[44] W. Lin, Z. Wu, L. Lin, A. Wen, and J. Li, "An ensemble random forest algorithm for insurance big data analysis," *IEEE Access*, vol. 5, pp. 16568–16575, 2017.

[45] A. Datta. *U.S. Health Insurance Dataset*. Accessed: Jul. 31, 2021. [Online]. Available: https://www.kaggle.com/teertha/ushealthinsurancedataset

[46] G. Anderson, "Random relational rules," Ph.D. dissertation, School Comput. Math. Sci., Univ. Waikato, Hamilton, New Zealand, 2008.

[47] Padraic. *Prudential Life Insurance Assessment*. Accessed: Aug. 5, 2021. [Online]. Available: https://www.kaggle.com/c/prudential-life-insurance-assessment/data

[48] E. Furman and R. Zitikis, "Weighted premium calculation principles," *Insurance: Math. Econ.*, vol. 42, no. 1, pp. 459–465, Feb. 2008.

[49] A. Heras, B. Balbas, and J. L. Vilar, "Conditional tail expectation and premium calculation," *ASTIN Bull., J. IAA*, vol. 42, no. 1, pp. 325–342, 2012.

[50] S. Glen. *Absolute Error & Mean Absolute Error, Statisticshowto.Com: Elementary Statistics for the Rest of U.S.!*. Accessed: Jul. 17, 2022. [Online]. Available: https://www.statisticshowto.com/absolute-error/

[51] A. Arora. *Quadratic Kappa Metric Explained in 5 Simple Steps*. Accessed: Jul. 17, 2022. [Online]. Available: https://www.kaggle.com/code/aroraaman/quadratic-kappa-metric-explained-in-5-simple-steps/notebook

[52] B. O. Soufiene, A. A. Bahattab, A. Trad, and H. Youssef, "PEERP: An priority-based energy-efficient routing protocol for reliable data transmission in healthcare using the IoT," *Proc. Comput. Sci.*, vol. 175, pp. 373–378, Jan. 2020.

[53] S. B. Othman, F. A. Almalki, C. Chakraborty, and H. Sakli, "Privacy-preserving aware data aggregation for IoT-based healthcare with green computing technologies," *Comput. Electr. Eng.*, vol. 101, Jul. 2022, Art. no. 108025.

[54] B. O. Soufiene, A. A. Bahattab, A. Trad, and H. Youssef, "LSDA: Lightweight secure data aggregation scheme in healthcare using IoT," in *Proc. 10th Int. Conf. Inf. Syst. Technol.*, 2020, pp. 1–4.

Anubhav Elhence received the B.E. degree in electronics and instrumentation engineering and the dual M.Sc. degree in physics from the Birla Institute of Technology and Science, Pilani (BITS Pilani), Pilani, India, in 2020, where he is currently pursuing the Ph.D. degree with the EEE Department.

He has worked as a Design Engineer with Xilinx, Inc., San Jose, CA, USA. His research interests include Internet of Things, IoT Security, Agriculture 4.0, and security in vehicular networks and drone networks.

Adit Goyal received the B.Tech. degree (Hons.) in computer science engineering from the Computer Science Department, Jaypee Institute of Information Technology (JIIT), Noida, India, in 2022. He is currently pursuing the M.S. degree in computer science with Northwestern University, Evanston, IL, USA.

His research interests include machine learning, data science, and quantum computing.

Vinay Chamola (Senior Member, IEEE) received the B.E. degree in electrical and electronics engineering and the master's degree in communication engineering from the Birla Institute of Technology and Science, Pilani (BITS Pilani), Pilani, India, in 2010 and 2013, respectively, and the Ph.D. degree in electrical and computer engineering from the National University of Singapore, Singapore, in 2016.

He is currently with the Department of Electrical and Electronics Engineering, BITS-Pilani, where he heads the Internet of Things Research Group/Laboratory. His research interests include IoT security, blockchain, UAVs, VANETs, 5G, and healthcare.

Dr. Chamola is a fellow of the IET. He serves as an Area Editor for the *IEEE Internet of Things Magazine* and *Ad Hoc Networks* Journal (Elsevier). He is also an Associate Editor of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE NETWORKING LETTERS, and the *IEEE Consumer Electronics Magazine*.

Biplab Sikdar (Senior Member, IEEE) received the B.Tech. degree in electronics and communication engineering from North Eastern Hill University, Shillong, India, in 1996, the M.Tech. degree in electrical engineering from the IIT Kanpur, Kanpur, India, in 1998, and the Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 2001.

He is currently an Associate Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. He serves as the Area Director of Communications and Networks Laboratory, NUS, Singapore. His current research interests include wireless network, and security for Internet of Things and cyber physical systems.