

Estd. 1964

Celebrating



BHARATI VIDYAPEETH



BHARATI VIDYAPEETH'S

Technical Magazine

2025 - 26

```
elif _operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
elif _operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

#selection at the end -add back the deselected mirror modifier object
mirror_ob.select= 1
modifier_ob.select=1
bpy.context.scene.objects.active = modifier_ob
print("Selected" + str(modifier_ob)) # modifier ob is the active ob
mirror_ob.select = 0
name = bpy.context.selected_objects[0]
#name.data ob path from scene object is
```

COMPILATION OF BEST RESEARCH PAPERS

Table of Content

Sr.No	Title of Research Paper
1	Machine Learning-Based Prediction of Drug Solubility in Multi-Component Drug–Solvent Systems Mr.Prathamesh A Gharat, Ms.Pranali R Palve, Dr. Suhasini Vijaykumar, Dr. Priya Chandran
2	AI Options Trader – Advanced Multi-Timeframe Ensemble Signal Engine Aniket Chougule, Suraj Jadhav
3	Cardio Care Analyzer: An Intelligent Machine Learning Framework for Early Cardiovascular Disease Risk Detection and Assessment Ms. Siddhika Belsare,Mr. Nilesh Mete,Ms. Shubhangi Mahadik
4	Neuroprompt : An Intelligent AI-Based Debugging Assistant for Automated Error Analysis and Resolution Dr. Priya Chandran ,Mayur Musale, Vaishakh Banawalkar
5	Blockchain-based Land Registry Framework Using Smart Contracts for Transparent and Secure Property Management Dr.Pratibha Deshmukh,Yogesh Gangarde

MACHINE LEARNING-BASED PREDICTION OF DRUG SOLUBILITY IN MULTI-COMPONENT DRUG-SOLVENT SYSTEMS

Pranali Palve¹, Dr. Priya Chandran², Prathamesh Gharat³ and Dr. Suhasini Vijaykumar⁴

¹Department of Computer Applications Bharati Vidyapeeth's Institute of Management and Information Technology Navi Mumbai, India

²Assistant Professor Department of Computer Applications Bharati Vidyapeeth's Institute of Management and Information Technology Navi Mumbai, India

³Department of Computer Applications Bharati Vidyapeeth's Institute of Management and Information Technology Navi Mumbai, India

⁴Principal, Bharati Vidyapeeth's Institute of Management and Information Technology Navi Mumbai, India

ABSTRACT

The physicochemical characteristic of solubility is one of the most critical properties affecting how well a pharmaceutical compound is absorbed, distributed and/or bioavailable. Therefore, accurately predicting the solubility helps support both drug discovery and the development of drugs/formulations. Drug solubility is most commonly determined using experimental methods which can be expensive and time-consuming for multiple drug/solvent combinations.

Recently developed machine-learning tools and chemoinformatic models provide an alternative way to predict physicochemical properties of chemicals using descriptors defined at the molecular level. In this study, we present the use of a new machine-learning framework to predict drugs' solubility in multi-component drug/solvent systems based on descriptor-based datasets.

Descriptors for training the models were obtained from multiple scientific sources, including Shodhganga research theses, PubChem, DrugBank, and ChemSpider. The training data was preprocessed using several techniques to create an appropriate structured dataset suitable for inputting into machine-learning algorithms. Preprocessing techniques included normalizing descriptors, handling missing values, and conducting feature engineering.

Three regression models—Ridge Regression, Random Forest Regressor, and XGBoost Regressor—were trained and evaluated using Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and coefficient of determination (R²). The results demonstrate that ensemble learning models provide reliable predictive performance for solubility estimation.

Index Terms—Drug Solubility Prediction, Machine Learning, Chemoinformatics, Random Forest, XGBoost, Pharmaceutical Modeling

I. INTRODUCTION

Overall, solubility is extremely important physicochemical characteristic that contributes to how a drug behaves in the body pharmacokinetically. How well a drug dissolves in a solvent will determine how quickly it will dissolve and absorb where it is needed, as well as how effective it will ultimately be as a therapeutic agent. Many times, poorly soluble drugs may have decreased bioavailability; therefore, evaluating how well a compound is soluble is crucial during the drug development and formulation phases.

Traditionally, solubility has been measured in the laboratory using experimental techniques such as equilibrium dissolution testing, spectrophotometric techniques, and chromatographic techniques, all of which can provide reliable data; but doing so takes significant amounts of time, requires specialized laboratory facilities, and requires multiple experimental trials for multiple drug-solvent combinations.

With recent advances in chemoinformatics and machine learning, researchers now have new opportunities to develop computational methods to predict solubility and other physicochemical properties of compounds. Machine learning can be used to build models that learn the complex relationships that exist between molecular descriptors and experimentally observed solubility, allowing researchers to estimate a drug's solubility characteristics before performing parts of the drug development process.

Besides the structure of the drug molecule itself, other factors such as the interactions occurring between solute and solvent molecules (e.g., hydrophobic interactions, hydrogen bonding, polarity compatibility, etc.) can significantly impact the ability of a drug molecule to dissolve in a given solvent, thereby determining the drug's overall solubility characteristics. It is important that predictive model development uses descriptors for both drug and solvent so that these properties can be factored into the prediction of drug solubility.

In this study we propose a machine-learning framework to estimate drug solubility based on molecular descriptors that have been using multiple well-reputed chemical repositories

i.e. Shodhganga, PubChem, DrugBank and ChemSpider. The collected descriptors from those repositories will be used to train multiple regression algorithms including Ridge Regression, Random Forest, and XGBoost. The model's effectiveness will be assessed using three well-documented regression metrics MAE, RMSE, and the coefficient of determination (R^2).

The primary contributions of this paper are summarized below :

- A machine learning framework to predict drug solubility for use in multi-component drug-solvent systems.
- Collection of molecular descriptor data from multiple reputable chemical repositories including Shodhganga, PubChem, DrugBank and ChemSpider.

- Implementation and comparison of three regression algorithms: Ridge Regression, Random Forest, and XGBoost.
- Evaluation of predictive performance using standard regression metrics including MAE, RMSE, and R2.

The remainder of this paper is organized as follows. Section II presents a review of existing literature on computational solubility prediction. Section III describes the system architecture and design. Section IV explains the data sources used for descriptor collection. Section V details the preprocessing and feature engineering steps. Section VI discusses the experimental setup. Section VII presents the model training process. Section VIII presents the results and analysis. Section

IX presents the descriptor contribution analysis. Section X discusses the findings. Finally, Section XI concludes the paper and outlines directions for future work.

The overall workflow of the proposed prediction framework is illustrated in Fig. 1.

II. LITERATURE REVIEW

Aqueous solubility prediction is a big area of research within chemoinformatics. Solubility is so important because it impacts drug absorption, bioavailability, and therefore formulation. Some of the early work in this area utilized various quantitative structure-property relationships (QSPR's) attempting to find the mathematical relationships between molecular descriptors and solubility, as measured in experiments. Delaney [1], developed the ESOL model - one of the most common methods for calculating aqueous solubility from molecular structure. QSPR models represent linear and therefore easy to interpret solutions; however, they do not adequately capture the complexity of nonlinear relationships between feature molecular descriptors and their resulting [physio]chemical characteristics.

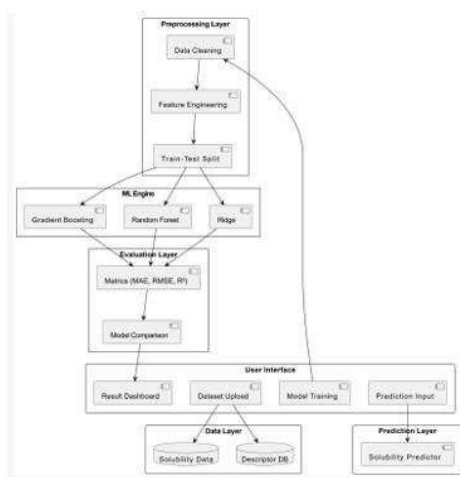


Fig. 1. Overall workflow of the machine learning based solubility prediction system.

To address these limitations, researchers began applying neural network techniques for modeling nonlinear patterns in chemical datasets. The work of Huuskone [2] demonstrated that using artificial neural networks can greatly enhance the accuracy of predictions for different types of organic compounds. With the rapid expansion of chemical databases and advances in computing power, the use of machine learning methods is becoming more common in chemoinformatics.

The use of ensemble learning algorithms for predicting molecular properties has yielded very good results. One great ensemble learning algorithm is Random Forest, first introduced by Breiman [4]. Random Forest models have gained popularity because they are able to effectively learn nonlinear relationships and are robust to noisy data about descriptors. The performance of the Random Forest model in this domain can be improved upon through the use of gradient boosting algorithms like XGBoost, which progressively add multiple weak learners to produce one strong prediction model [3].

In addition, recent research has shown that deep neural networks can be utilized for predicting chemical properties (e.g., aqueous solubility). When trained with large datasets, Lusci et al [5]. found that deep neural networks are more accurate than former quantitative structure activity relationship (QSAR) models. Due to the high volume of data and computational power required to train these models, they are generally slower than the older QSAR methods.

A major part of the research examining how soluble something is will be based on using curated chemical datasets. An example of this is when Sorkun et al [6]. produced AqSolDB which includes data for many thousands of compounds' solubility (the solubility of a chemical in a given solvent) and their molecular descriptors. The existence of these datasets has led to the systematic benchmarking of solubility predicting machine learning algorithms against data from these datasets. For our study, we acquired molecular descriptors from available databases on the web and/or online software programs such as PaDEL [7]. that allow for building molecular descriptors.

Although scientists have made considerable advancements in our knowledge of solubility, predicting solubility in complex mixtures of numerous pharmaceuticals in many cases is still difficult. While approaches to understanding solubility based on advanced descriptors and machine learning have made significant advances, complex solute-solvent interactions (e.g., hydrogen bonding, hydrophobic interactions, and polarity of the solute and/or solvent) do not always correlate with simple descriptors.

There have been multiple studies that demonstrated promising approaches to improving the accuracy of predictive solubility models based on the use of advanced descriptors. For example, Tetko et al [8]. demonstrated how consensus models enhance prediction accuracy. Wu et al. [9] proposed using deep learning architectures to predict chemical properties. Kearnes et al. [10] provided an example of how the use of graph convolutions can be used to represent molecular structures.

Additional studies from Mayr et al. [11] and Sheridan [12] provided evidence that descriptive feature representation and descriptor selection are crucial for improving the predictive power of models. Research from Goh et al. [13] demonstrated the use of chemoinformatics approaches based on neural networks to predict chemical properties. Additionally, Liu et al. [14] studied the use of hybrid machine learning from chemical property prediction.

The results of these studies suggest that by using multiple descriptors and advanced machine learning techniques, solubility prediction accuracy can be dramatically increased.

III. SYSTEM DESIGN

The proposed architecture for the Solubility Prediction System will allow for processing of datasets, model training and prediction of solubility for drug and solvent combinations. The user/system interactions with the various elements of the architecture are illustrated in the diagram as follows; 2.

This particular system will be used primarily by researchers who will upload their molecular descriptor datasets into the system, train their model and receive solubility predictions using the resulting trained model. The uploaded dataset will be processed by the system through a number of stages including preprocessing, feature extraction, training a model, and evaluating the performance of that model.

Once the model has been trained, it can be used to predict the solubility of new drug/solvent combinations from their corresponding molecular descriptors. This workflow will allow for evaluation of potentially suitable solvent systems without requiring extensive laboratory experimentation by researchers.

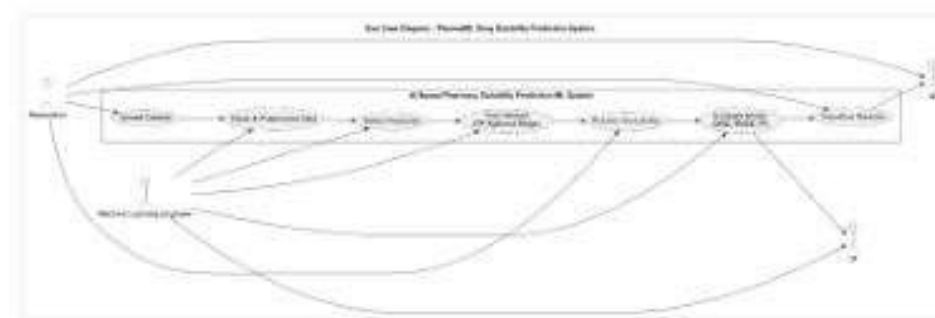


Fig. 2. Use case diagram illustrating the interaction between the researcher and the AI-based solubility prediction system.

IV. DATA SOURCES AND DESCRIPTOR COLLECTION

This research makes use of a dataset developed using molecular descriptors that were obtained from many well-established chemical databases to gather realistic chemical information about pharmaceutical compounds.

PubChem, which is under the jurisdiction of the National Center for Biotechnology Information (NCBI) is a huge open-access repository of chemicals. There are over millions of compounds (which include chemical structure, molecular properties, biological activities, as well as physicochemical properties) with the chemicals from PubChem being one of the primary sources of the data used to extract molecular descriptors for both drug substances (molecules) and for solvents.

Another commonly utilized chemical database is DrugBank, which contains extensive information (including but not limited to molecular structures with property descriptions, pharmacological data, and drug-target interactions) that incorporate data about drugs. The descriptors found in DrugBank can provide physicochemical characteristics of drug substances and facilitate identification of characteristics of the drug substance that affects solubility.

Additional descriptor information contained in datasets from an additional chemical database, ChemSpider, has been aggregated from many sources and the chemical structures and properties have each been validated through curation by the Royal Society of Chemistry or other authoritative sources. The primary application of ChemSpider-specific descriptor data was to further validate and augment descriptors collected from the other chemical databases that were used.

In addition to these chemical databases, research on solubility behaviour and physicochemical analysis of pharmaceutical compounds found in Shodhganga (a digital repository) was also reviewed. Shodhganga contains postgraduate and doctoral theses from Indian universities, many of which have conducted extensive studies on drug-solvent interactions and pharmaceutical formulation analysis.

Molecular descriptors for both drugs and solvents were extracted from both repositories. The selected descriptors include molecular weight, topological polar surface area (TPSA), hydrogen bond donor and acceptor counts, rotatable bond count, and partition coefficient (LogP). These physicochemical descriptors have common use in chemoinformatics as they predict an important structural feature that comprises the solubility behaviour.

Descriptor data was compiled into a structured dataset that represented drug-solvent pairs and corresponding molecular properties; it was used as input in order to train and evaluate machine learning models.

V. METHODOLOGY

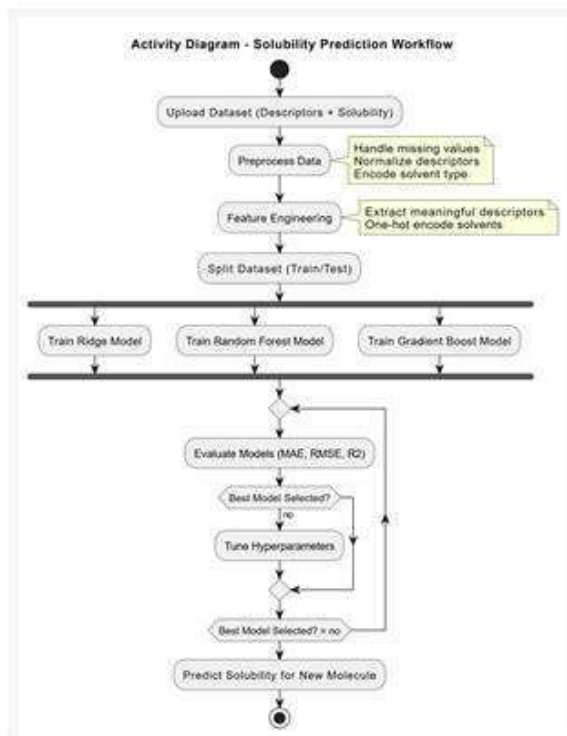


Fig. 3. Activity diagram illustrating the workflow of data preprocessing and feature engineering.

The proposed framework follows a structured workflow consisting of data preprocessing, feature engineering, machine learning model training, and performance evaluation. The methodological workflow adopted in this study is illustrated in Fig. 3.

A. Data Preprocessing

Inconsistencies in datasets obtained from various chemical sources may have detrimental effects on the effectiveness of machine-learning algorithms and must be corrected prior to training. Duplicate entries must be identified and removed from the dataset to avoid introducing bias into the learning process. Clean and standard numerical representations must be produced from descriptor values containing non-numeric symbols or estimated values.

Missing values must be filled in using the median as an estimate because it produces the least amount of distortion to the overall statistical distribution of the dataset. Once

Normalisation will prevent descriptor values with larger magnitudes from dominating the model-training process and will allow equal contributions from all descriptors during model-learning.

B. Feature Engineering

To better represent interactions between drug and solvent molecules, additional interaction-based features were generated from the existing molecular descriptors

$$\text{LogP}_{\text{difference}} = \text{LogP}_{\text{drug}} - \text{LogP}_{\text{solvent}} \quad (1)$$

$$\text{TPSA}_{\text{ratio}} = \frac{\text{TPSA}_{\text{drug}}}{\text{TPSA}_{\text{solvent}}} \quad (2)$$

The difference in LogP gives a measure of hydrophobic compatibility between solvent and drug molecules, while the ratio of total polar surface area (TPSA) is a measure of polar interactions affecting the dissolution behaviour. Incorporation of these descriptors based on the interactions between drugs and solvents allows machine learning models to more accurately reflect the physicochemical relationships that describe the solubility of pharmaceuticals.

C. Implementation Workflow

The implementation workflow of the proposed system is illustrated in Fig. 4. This workflow represents how the prepared dataset flows through different stages of processing and model execution.

The sequence diagram demonstrates how a dataset is received, and molecular descriptor databases are loaded. The system checks if there are any missing values, and cleans and imputes, if applicable. The dataset is split into training and testing phases after it has been pre-processed. Feature selection and normalisation will be applied before multiple machine learning algorithms are trained. The final model will be selected based on evaluation metrics, to predict solubility values. The sequence diagram provides a clear visual representation of the movement of data through each phase of the workflow.

VI. EXPERIMENTAL SETUP

To test how well a computer can predict how well a drug will dissolve, we used multiple dimensionality of molecular descriptors for each of these drugs and solvents.

The data is made up of several hundred combinations of solvent/drug pairs that have been described in numerous papers, e.g. PubChem, DrugBank, ChemSpider, and cited in Shodhganga.

The dataset developed was divided into a training dataset and a test dataset at a rate of 80

During the training of the models, feature scaling and normalization was applied so that all descriptors could contribute equally during the training process, while descriptors with larger numerical ranges could have potentially dominated the training process and caused the models to have poor performance.

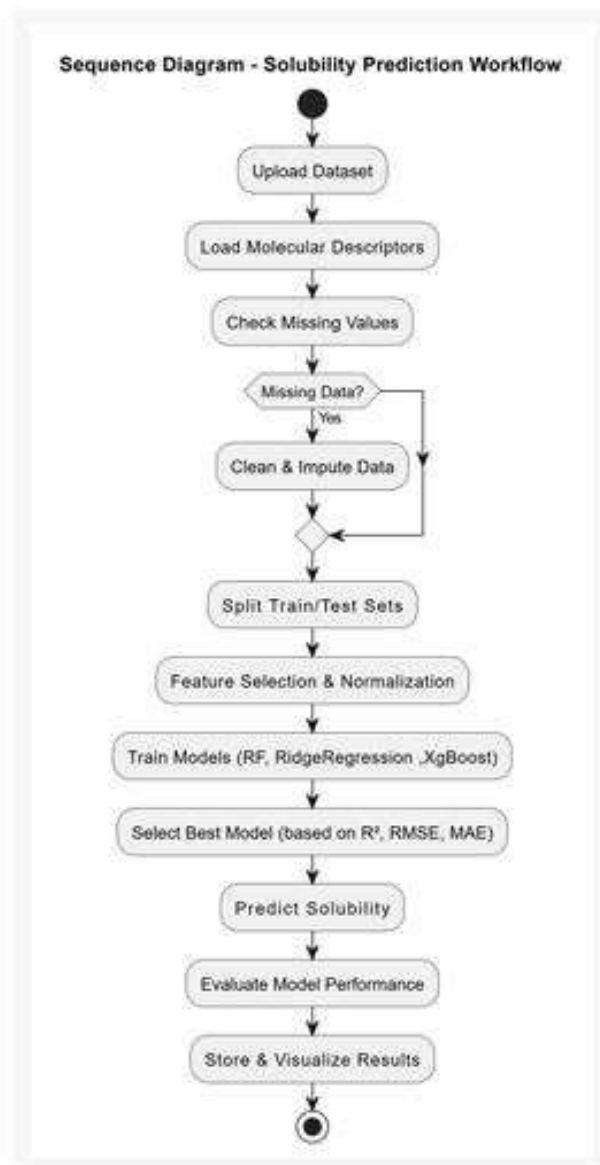


Fig. 4. Sequence diagram illustrating the workflow of data processing, model training, and solubility prediction.

To determine the settings for the various machine learning algorithms, hyperparameter tuning was conducted using grid search optimization. For trains of the Random Forest model, parameters that were tweaked were the number of trees and maximum tree depth. For the XGBoost model, parameters that were evaluated were learning rate, tree depth and number of estimators to determine which were the optimal settings for the model.

The performance of each model was assessed through three standard regression metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Coefficient of Determination (R^2). These metrics provide quantitative measures of a model's prediction accuracy and ability to generalize from training data.

VII. MODEL TRAINING

Three machine learning regression algorithms were implemented for predicting drug solubility.

A. Ridge Regression

The Ridge Regression modelling technique adds L2 Regularisation to linear regression for regularization to decrease overfitting and improve stability of the model. Ridge Regression was utilized in this study as a baseline model in order to evaluate the degree to which linear relationships exist between the molecular descriptors and the solubility values.

B. Random Forest

In contrast to Ridge Regression, Random Forest uses an ensemble learning approach, where multiple (many) decision trees are built and used to combine their predictions ultimately generating the final output. The use of multiple decision trees increases the stability of the prediction and reduces the potential for overfitting. Random Forest does an excellent job of capturing nonlinear relationships between molecular descriptors.

C. XGBoost

XGBoost is a type of Gradient Boosted Decision Tree model that is built sequentially to minimize error from predictions by creating a new tree to fix the current prediction error from the previous model. As a result, XGBoost is widely used in machine learning for quickly processing complex datasets.

VIII. RESULTS AND ANALYSIS

Model performance was evaluated using three standard regression metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and coefficient of determination (R^2). These metrics provide quantitative insight into prediction accuracy and the ability of the models to generalize across unseen data.

TABLE I MODEL PERFORMANCE COMPARISON

Model	RMSE	MAE	R ²
Ridge Regression	1.1884	30.12	-0.1638
Random Forest	0.9480	24.95	0.1383
XGBoost	0.9883	22.85	0.3532

Table I The evaluated machine learning models' performance has been summarized. The Ridge Regression model provided the least performance, indicating that the modeling of drug-solvent solubility behaviour cannot be accurately modelled solely using linear relationships. The Random Forest model produced the lowest RMSE value which illustrates strong robustness for capturing the non-linear interactions of molecular descriptors. The XGBoost model R² provided the highest score, thus demonstrating superior ability to explain the degree of variance in solubility data.

Fig. 5 has a comparative analysis of the MAE of the Ridge

Regression, Random Forest, and XGBoost models. MAE is an important measure because it shows how large (on average) the errors in the predicted values are however it does not measure whether they are above or below the actual value. From the graph you can see that the XGBoost model had the lowest MAE which means it was more accurate and consistent at predicting solubility values than both the Random Forest Model and the Ridge Regression Model. The Random

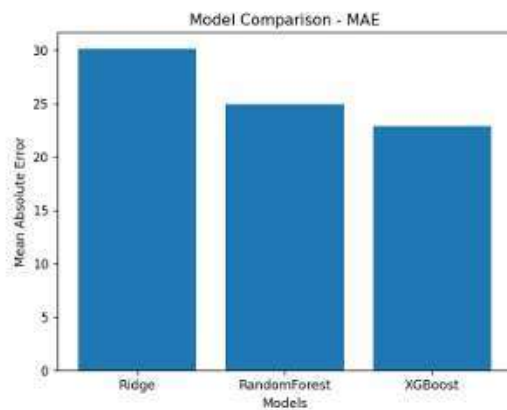


Fig.5 MAE comparison of the evaluated machine learning models.

Forest Model performed moderately in terms of accuracy and consistency; it scored lower with MAE than Ridge Regression but does not score as low with MAE as XGBoost. The Ridge Regression model had the highest MAE which shows that it was unable to accurately model the complex nonlinear relationship that exists between the solvent and drug solubility data.

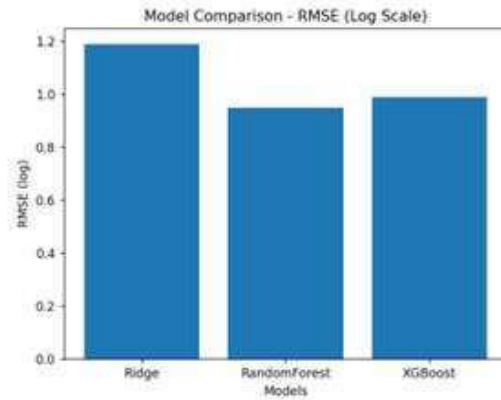


Fig.6. RMSE comparison of machine learning models on a logarithmic scale.

Fig. 6 The Root Mean Square Error (RMSE) comparison between models is represented on a logarithmic scale. The RMSE is significant because it penalises larger errors more heavily. Thus, using it to assess models, its application is appropriate in cases where there are large deviations from expected results.

The Random Forest model has the lowest RMSE value, suggesting that it effectively reduces large prediction errors and handles data variability. The next lowest value belongs to the XGBoost model, just slightly higher than that of the Random Forest; however, it has demonstrated significantly better performance compared to Ridge Regression.

Additionally, Ridge Regression has the highest RMSE value of any of the models evaluated, confirming that it has limited ability to model nonlinear relationships present in the data.

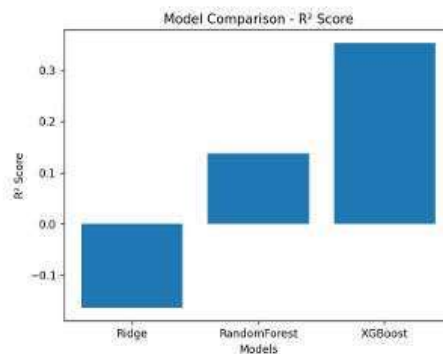


Fig.7. R2 score comparison across the evaluated machine learning models.

Fig. 7 The figure presents the R2 value of the three models. R2 is a measure of how well the fit of the model describes the variance around the target. XGBoost had the highest R2, indicating that it captures the underlying correlations in the dataset, thus allowing for the most accurate predictions of the three models. Random Forest performed at a moderate level with a positive R2, which suggests that it has some predictive ability. In contrast, Ridge Regression produced an R2, which indicates performance below the simple mean, and therefore is not appropriate for this problem domain.

IX. DESCRIPTOR CONTRIBUTION ANALYSIS

It is beneficial to know which molecular descriptors have the most significant effect on solubility predictions because they can help future researchers and developers understand how to interpret machine learning model outputs. In this context, descriptor contribution analysis aids researchers and developers in determining the physicochemical descriptors that largely affect model outputs.

Random Forest is an excellent tool for determining the relative importance of descriptors within a data set, as it estimates the importance value for each descriptor based on its ability to reduce error in prediction accuracy (i.e., its ability to "split" or categorize decision trees).

The analysis results show that the two principal types of descriptors that affect model output are polar and hydrophobic descriptors. Results indicate that logP and topological polar surface area (TPSA) descriptors have a pivotal effect on solubility predictions by having a direct correlation to inter- molecular attractions between solute and solvent molecules.

Creating interaction descriptors like logP difference and TPSA ratio enabled the model to predict solubility better by correlating the solubility of the drug to that of the solvent, thus allowing for a more realistic representation of the physicochemical relationship between the solubility of the solute and the characteristics of the solvent.

These results are consistent with the basic principles of pharmaceutical chemistry, where molecular polarity, hydrophobic intermolecular attraction, and ability to form hydrogen bonds in the solubility of compounds are important aspects in determining the solubility of a compound.

It can be observed from the results of this study that choosing suitable molecular descriptors has a major impact on how accurate the models produced were. Thus, the influence of both Polarity (TPSA) and Hydrophobicity (LogP) indicates that both characteristics are important in determining drug solubility in regards to solvent systems.

X. DISCUSSION

According to the results of the experiment, ensemble-based methods have outperformed the use of a simple linear regression method as a prediction model when using these methods to predict drug solubility in one or more solvent systems.

The Random Forest ensemble method exhibited stable predictive accuracies when applied to the various drug/solvent combinations and is able to learn to model non-linear relationships between molecular descriptors effectively.

XGBoost also performed competitively to Random Forest in minimizing the average prediction error; however, Random Forest provides a higher level of interpretability when compared to the other algorithms, due to Random Forest's inbuilt ability to conduct a feature importance analysis, which allows researchers to identify molecular descriptors contributing most to the solubility of a compound.

Utilizing molecular descriptor data from multiple chemical databases increased the diversity of the datasets and improved the generalizability of all models. The combination of descriptor data from the PubChem, DrugBank, ChemSpider and Shodhganga datasets enabled the creation of a robust representation of drug/solvent molecular systems.

The utilization of machine learning solubility prediction frameworks in support of pharmaceutical research give researchers the ability to screen multiple candidate molecules and solvent systems quickly and efficiently, prior to conducting laboratory experiments. The use of these computational tools help reduce the time spent on the development of the drugs, as well as assist with the design of drug formulations in the earlier stages of the drug development process.

XI. CONCLUSION

In this research, we developed a machine learning model that predicts the solubility of drugs in multiple solvents (multi-component drug; solvent systems) based on molecular descriptors obtained from Shodhganga, PubChem, DrugBank and ChemSpider.

The prediction results were produced by implementing ensemble learning algorithms, specifically Random Forest and XGBoost; these methods provided dependable predictions based on the solubility behaviours of the all the molecular descriptors.

This research also illustrates the ability of machine learning methods to aid in pharmaceutical research by allowing for faster and more efficient computational screening of drug candidates and solvent systems.

Future work could include additional molecular descriptors as well as exploring deep learning methods in order to increase the accuracy of the predictions and generalisation of the models.

REFERENCES

- [1] J. S. Delaney, "ESOL: Estimating aqueous solubility directly from molecular structure," *Journal of Chemical Information and Computer Sciences*, 2004.
- [2] J. Huuskonen, "Neural network modeling for solubility prediction," *Journal of Chemical Information and Computer Sciences*, 2000.

-
-
- [3] T. Chen and C. Guestrin, "XGBoost," KDD, 2016.
 - [4] L. Breiman, "Random forests," Machine Learning, 2001.
 - [5] A. Lusci et al., "Deep learning for solubility prediction," Journal of Chemical Information and Modeling, 2013.
 - [6] M. Sorkun et al., "AqSolDB dataset," Scientific Data, 2019.
 - [7] C. W. Yap, "PaDEL descriptor," Journal of Computational Chemistry, 2011.
 - [8] I. Tetko et al., "Virtual computational chemistry laboratory," J. Comput.-Aided Mol. Des., 2005.
 - [9] Z. Wu et al., "MoleculeNet: benchmark for molecular ML," Chemical Science, 2018.
 - [10] S. Kearnes et al., "Graph convolutional networks for molecular data," J. Comput.-Aided Mol. Des., 2016.
 - [11] A. Mayr et al., "DeepTox: toxicity prediction," Frontiers in Environmental Science, 2016.
 - [12] R. Sheridan, "Interpretation of QSAR models," J. Chem. Inf. Model., 2013.
 - [13] G. Goh et al., "Deep learning for computational chemistry," J. Comput. Chem., 2017.
 - [14] Y. Liu et al., "Hybrid ML for property prediction," Chemometrics, 2020.
 - [15] D. Rogers and M. Hahn, "Extended-connectivity fingerprints," J. Chem. Inf. Model., 2010.

AI OPTIONS TRADER – ADVANCED MULTI-TIMEFRAME ENSEMBLE SIGNAL ENGINE

¹Prof. Sudeshna Roy, ²Aniket Prakash Chougule and ³Suraj Jadhav

¹Master of Computer Application (MCA), Bharati Vidyapeeth's Institute of
Management and Information Technology, CBD Belapur, Navi Mumbai

²Master of Computer Applications (MCA), Bharati Vidyapeeth's Institute of
Management and Information Technology, CBD Belapur, Navi Mumbai

³Master of Computer Applications (MCA), ³Bharati Vidyapeeth's Institute of
Management and Information Technology, CBD Belapur, Navi Mumbai

ABSTRACT

Cryptocurrency trading is considered as one of the tough and little complicated field. It is open all the time, it does not have any central regulator. Therefore it is very different from other financial markets. because of that the prices of cryptocurrencies changes very quickly and cannot be predictable. Resulting doing trading in cryptocurrencies manually can be very challenging, and also does not work a lot.

Relative strength index and moving average convergence divergence this factors are mainly used in making traditional trading strategies. They are helpful but not that much good for dealing with such sudden changes in the market. They give signals but those may be confusing for traders.

Nowadays, Automated trading systems have become more popular with the help of Artificial intelligence and Machine learning. This systems find patterns also make predictions regarding future price movements after studying existing data. even so many of these systems acts like boxes resulting in very complicated for users to learn the process of make their predictions. As well they do not have strategies for risk management, which have impacts on practical use.

This research tells about a type of trading system which combines factors like machine learning models along with technical validation & management of risk. The objective is to create transparent system, adapting changing conditions of markets and also predicting prices.

Analysis techniques such as Relative strength index, Moving average convergence divergence and Exponential moving averages. Some more factors it uses like candlestick pattern recognition and fair value gap detection. Risk management which based on average true range helps user in adjusting stop-loss and take-profit regarding market volatility.

System having step-by-step process starts from collecting data preprocessing, training models then validating them and last generating signals. Results of Experiment tells that combining learning along with multi-timeframe confirmation resulting reduction in false signals & also helps in improving trading performance. The system has features like more consistent, reliable and having risk-reward advantage than that of traditional trading.

The cryptography market is fast growing also changed the style of people's trading. It is open for 24/7 means all the time available, also highly volatile and always active market. Therefore while trading traders cannot depends on human strategies because of price volatility. So, there is need of automated trading systems that can handles such complexity.

INTRODUCTION

Traditional trading approaches are based on indicators, like the Relative Strength Index, Moving Average Convergence Divergence and moving averages. While these tools are useful they have limitations when it comes to dealing with the cryptocurrency market. Cryptocurrency trading systems need to be able to adapt to the nature of the market. Cryptocurrency markets are constantly. Cryptocurrency trading systems need to be able to keep up with these changes. The cryptocurrency market is a field and cryptocurrency trading systems need to be able to handle this challenge. trading systems capable of handling such complexity.

Conventional trading methods mainly use indicators like the Relative Strength Index (RSI) Moving Average Convergence Divergence (MACD) and moving averages. These methods are used by trading systems that can handle complexity.

Conventional trading methods mainly use indicators like the Relative Strength Index (RSI) Moving Average Convergence Divergence (MACD) and moving averages. These indicators help us understand market trends and momentum. They usually work alone and only look at one timeframe. In volatile markets they can give wrong or mixed signals.

The use of Artificial intelligence and Machine learning in markets becoming very popular. Such technologies help us to analyze lot of real time data. They research patterns and relations so it is hard to see along with traditional methods. The ML models such as random forest & gradient boosting resulting very good at predicting outputs. Many of machine learning based trading systems having a lot of limitations.

Traditional trading strategies consistently use analysis indicators such as relative strength index. Moving average convergence divergence and Exponential moving averages are mostly used for finding trends and also entry exit points. Because of simplicity and ease of understanding makes these indicators more popular. They are not good for high volatile markets.

They depends on predefined rules and single time frame analysis. This reduces their ability to transform to changing market situations. Traders repeatedly get signals also delayed reactions & inconsistent outputs.

Integration of artificial intelligence and machine learning into trading systems is the only solution to overcome from these issues. This model can ability to process lot of data then find patterns and also make predictions. Algorithms such as random forest, gradient boosting and neural networks work better in predictions tasks.

Many of the systems are complex to learn. They acts as black box models. This makes it difficult for users to trust on their decisions. This thing reduces confidence of user. Also making it complex to use such systems in real world trading.

Background

Since last ten years Algorithmic trading has changed a lot. Traditional trading systems used some rules including one is when to buy or sell at moving averages another is stock was oversold.

They helpless in unpredictable markets. Then machine learning came in. Helps systems to go beyond this simple rules. For analyze data, techniques such as decision trees, support vector machines and random forests were helped.

These methods helps to make predictions with more accuracy. But these methods comes with new problems like overfitting and it was very complicated to learn why they made such predictions. Then deep learning models such as long short-term memory networks progressed. It did an even good job at learning data overtime. These models need lot of data also powerful computers and along with that it's still difficult to understand how they make their forecasts. Coming with all these advantage lot of trading systems still how predicting & checking and managing risk as diff things. This way does not work good because its all dividing.

So people need a system that mix-up making predictions with checking & controlling risks. Another big challenge in forecasting markets is that its all uncorrected and unpredictable. How people feel about the market how much money is available & how much quantity is being traded and unexpected events are some things directly impacts on momentum of prices.

Some old statistical models fight with all these relationships, which limits how good the can forecast. Machine learning though is better at looking patterns and connections in variables, which making it better, for analyzing existing financial data. Where patterns are not always in flow.

In spite of their advantages machine learning models wants designed features to work good. Raw market data is not that well to get insights. Therefore feature engineering might be important in process of turning data into useful information that makes models work well. The technical indicators which made from price and volume data give insights by studying volatility, momentum and trends.

Machine learning models have ability to learn market behavior even better when we add price action things such as candlestick patterns and structure of market. This things helps model to learn market behavior even good than human does. Two keys to getting insights from market are machine learning models & feature engineering.

For making machine learning model, feature engineering and technical indicators are really important factors. Making sure the signals are good enough is another important thing. When getting signals for short time periods they can be incorrect because of scenarios happening for time.

This results in make the outputs not reliable. On other hand stalking at longer time periods gives you good view of what exactly happening right now in the market but it can be difficult to find a good entry point of market.

Use of time periods helping a lot to traders because they able to view both big picture and small details. This path gave opportunity for match up long term trends with short term chances. People think that this is a path of making trading accurate but it is not actually used in automated systems.

Moreover trying to forecast what exactly happen and making assurance about signals are better a trading system also needs to have ability to manage expected risk. Even the better forecasting models can make mistakes. If you are not able to manage risk you can lose a lot of your money in few trades. Rather than using fixed plan it is better way to do things in live markets, making strategies is one of the best risk management theory. For making a good trading system feature engineering, technical indicators with validation as well as risk management are very much important.

OBJECTIVES

The first and main goal of research is to build a trading system. Using machine learning techniques to make sure the signals of trading are reliable and also a consistent towards time intervals. Meaning of this is we shall look at trends of market on various time frames. Examples like 1 hr., 15 min, 5 min charts.

Indicators such as RSI, EMA and MACD are good sense of markets trends that we want to use. In addition to that we go through the price action patterns, like candlestick for learn about market better. Trading system shall using Average true range for risk management. Meaning that system will adjusting stop loss and also take profit levels depends on volatility of market.

One more goal is to reduction in trading signals. We will achieve this by using validation layers, that includes machine learning predictions & technical confirmations. This helps the system make more correct predictions and also improving entire performance. For understanding how the system works the trading system keep logs of trading decisions.

This research targeted to making trading system reliable and more transparent. We want users can trust our system and learn the reason behind its given decisions. Using combine factors like indicators & machine learning techniques system providing clear directions into operations. This helping a lot to users for making trading decisions also improving entire performance of system.

Creation of system that is dynamic and also adaptable is our main focus. After responding changing market situations with adjustment of its risk management art as needed. This leading to help protect capital and also maintain a risk reward proportion. Entire the goal of research is to build trading system that is more correct, transparent and also reliable.

To get assurance of trading decisions are enough consistent across various time frames the trading system shall use timeframe validation. This way will help figure out noise and reduction in similarities of signals. System will be able to create informed decisions after analyzing markets trends on various time bounds.

To improve quality of forecasting we have option to use a set of technical indicators. These helps to catch momentum and also trend direction giving insights in market situation. After combining these indicators along with price action techniques for generation of signals that are enough accurate and reliable.

Main motive behind research is to improve trading system efficiency. The system will enhance performance and consistency in trading by reducing signals of trading and improvement of signal quality. The final milestone is to build trading system that is more reliable, transparent & accurate that gives users with information they needed to create good trading decisions.

LITERATURE REVIEW

In last few years the use of machine learning in markets has been studied a lot. It is very true when we trying to find out prices movements and when to buy or sell. At very first people used methods such as linear regression, decision tree and support vector machine. These ML methods were better at figure out patterns in price data structure. It did a good job of forecasting prices than that of traditional methods.

In many cases they were not good at facing with changes in market and also prices that was not follow a line, which is more common when you trades in cryptography.

People starts with use methods which is combined of ML models such as random forests & gradient boosting. Helping to make predictions accurate and more reliable. These ML models works by combine effects of results of models, results in reduction of chance of getting it incorrect and also makes create predictions more common. Because of having ability to handling noisy data and complex relations between various items these ML models are better at forecasting prices in markets and this shown in research.

Along with this advantages many of trading systems which uses machine learning are still have not enough transparency which making it difficult for peoples to learn how they build predictions and also trust them while making trades.

Machine learning is not only the path to make trade decisions but also look over the charts and trends. To find trends means see how fast prices were moving and find out when prices may be change direction peoples use things such as RSI, moving averages and MACD. Complex methods such as stalking candlestick patterns and also price movements can give little more insight into what actually market doing now.

Depending on these kind of methods may lead to inconsistent results because they might be give false of little conflicting signals and that factor can lead confusion for

people who trying to build trading decisions depending upon ML and technical analysis of many financial markets and also cryptocurrency trading.

Depending upon these advantages many systems which use ML are still not clear which makes it difficult for people to learn how they create predictions and also trusting on them when doing trades.

People who depending on methods to create decisions can getting incorrect results. This is only because of these methods consistently give signals. It can be getting confusing for people who trying to create decisions based on ML and cryptocurrency trading. Especially it is very complicated when market is over to the place.

Systems required a lot of data and little powerful computers. Also it is difficult to learn how they create decisions, which makes them little less suitable for trading. In trading it is really important to understand how decisions are made.

Another important thing is managing risk. Many trading systems use fixed stop-loss and take-profit levels. These levels do not change when the market changes. Studies have shown that using volatility based risk management techniques is important. These techniques use the Average True Range to adjust trading parameters.

These approaches have been shown to make trades last longer and reduce losses.. Combining dynamic risk management strategies with machine learning based prediction systems is still not common.

PROBLEM STATEMENT

Cryptocurrency trading is very tough. This is because it is highly volatile and happens all the time. There are also price swings. Traditional trading approaches use indicators like RSI, MACD and moving averages. These indicators give buy and sell signals.. These indicators often work alone. They are not good at adapting to market changes. As a result they often give signals. This is especially true in market conditions. It leads to trading decisions and financial losses.

With more people using machine learning in markets predictive models have been introduced to improve trading accuracy by looking at old data and finding complex patterns. While these models are better than methods they also introduce new challenges. Many machine learning models are being used in cryptocurrency trading and markets. Cryptocurrency trading and markets are using machine learning to make predictions. Machine learning and technical analysis of markets and cryptocurrency trading are being used together to make trading decisions. The market and cryptocurrency trading are very important, to people who are using machine learning and technical analysis to make decisions.

Systems work like a box that you cannot see inside they make predictions. Do not explain how they got those predictions, which makes users not trust them very much and they are not very useful in real life. Also these systems are usually trained with data and they have a hard time adjusting to changes in the market that happen very quickly which makes them not work well when things are happening in real time.

Another big problem with systems that exist now is that they do not handle risk well. A lot of systems use strategies to decide when to stop a trade or take a profit but these strategies do not take into account how much the market is changing. In markets like the one for cryptocurrency which's very unpredictable using these simple strategies can cause trades to end too soon or result in big losses. Because systems do not have a way to check if their signals are correct the signals they produce are not very reliable.

In addition to these problems systems that use machine learning and traditional methods of analyzing the market are not usually connected. Each method has its strengths but they are often used separately which means they do not work as well as they could. There needs to be a system that combines machine learning, technical analysis and risk management into one framework. This system should not make better predictions but also provide results that make sense and work well in different market conditions.

Machine learning methods have been used to try to fix some of these problems by allowing systems to find patterns in data and make predictions based on that data. Trading systems like these can learn from cryptocurrency markets and other markets and trading systems can make data-driven predictions, which's a big improvement over systems that do not use machine learning.

Using these approaches having made predictions very accurate. In many cases they also have some issues. Many ML models were such as boxes, they do not have ability to explain how they create predictions. This making it difficult for many users to trust them.

It is also complex to check if the signals which they are giving are reliable. Machine learning models can also have sensitive modifications in market. They may not able to work with data. One more big problem with systems is that they do not having any validation methods. Signals getting from one timeframe may not able to show entire market trend.

They may be affected by a short term volatility. These signals can lead to very bad decisions without checking timeframes. Many of systems do not combine the study with machine learning forecasting. This can leads to limits their abilities to use both things domain knowledge and date insights. finally results in they failed to develop trading strategy.

Management of risk is also one of concern. Many of solutions come up with the use of stop loss and take profit strategies. These can not be adapting changing market conditions. These can lead to early exits or big losses. The lacking in risk control gets making trading system less effective. It increases chance of outputs.

The system must give clear explanations and provide predictions. Traders always need a signals. Without giving any explanation is it difficult to trust on automated trading system. This clearly shows need of a system which balances accuracy as well as transparency.

It must be good working in market situations. Therefore main problem is that this research is addresses is a creating system which is integrated as well as smart & explainable. It providing signals by combining various methods. After every prediction there must be explanations. It also has ability to adapt market situations. Traders have to trust on signals it gives. System improves trading results with helping a lot to them.

METHODOLOGY

The system proposed is very simple and using machine learning and technical analysis for generation of trading signals for who going to trade. it starts by fetching real time market data from the official cryptocurrency exchange API. That data including very high low closing prices with trading volume. After that this raw data is going to cleaning up which meaning is handling missing values and removing incorrectness and then transforming data into a format which is easier to analyze. The system looking for information which can help improve the model once the data is clean. Strength index and moving average convergence divergence this kind of indicators system uses. Also it uses stuffs such as candlestick recognition & fair value gap detection for capture prices action behavior.

After that data is labeled into three parts which are buy, sell & hold based on price actions. For forecasting what things will happen system uses this ML techniques. It trains models such as random forest and gradient boosting on data. After combining their outcomes. This going to help to reduce overfitting and also making system accurate. System checks predictions across time intervals such as 1 hr., 15 min, 5 min. This gives assurance that trading signals were consistent and reduced impact of noise. The system also has feature of risk management mechanism depending upon average true range. It adjusting stop loss and take profit levels regarding to market volatility, which is trading signal system. For proposed system the trading signal system is really important. It uses it to make predictions.

SYSTEM ARCHITECTURE

Trading signal system is about how it uses machine learning for generating trading signals for system. The system helps trade adjusting to change conditions and improving risk control. This also helps to keep record of all decisions it makes.

It is really helping for making it easier to check how good trading signal system working. The system is combination of machine learning, risk management and technical analysis. This makes it good and easy for trading. Multiple timeframe helping to analyze both short term and long term market trends. The data which is collected is consistently updated at regular time intervals, and enables system to operate in real time situations.

Next layer consisting data preprocessing and also feature engineering section, in which raw data going to transform into well-structured format comfortable for machine learning.

This stage also including cleaning dataset by handling the missing values also removing incorrectness and standardizing data. Above preprocessing and feature extraction is important.

It brings together data processing, machine learning, validation and risk management into one workflow. The architecture starts with a data collection part. This part gathers time and historical market data from a cryptocurrency exchange API. The data includes opening, high, low and closing prices, trading volume and timestamps. This data is the base for analysis. The collected data goes to a preprocessing part.

Here it is cleaned, normalized and changed into information using indicators like RSI, MACD, EMA and ATR. Other analysis tools, such, as recognizing candlestick patterns and detecting Fair Value Gaps (FVG) are also used to make the data more valuable. The AI-based trading system uses machine learning models.

These models are a combination of models. The AI-based trading system uses these models to make trading decisions. The machine learning module uses the processed data. This data helps the AI based trading system to learn and make predictions. The AI-based trading system relies on machine learning. Machine learning helps the system to make trading decisions.

The data is then sent to the machine learning part of the system. This is the core of the system. It uses a method called learning. This means it uses models like Random Forest and Gradient Boosting. Each model looks at the data.

To get assurance the predictions are well system checks them a times. It looks over the predictions, in time frames like 1 hr., 15 min & 5 min. This thing helps to get rid of forecasting not strong. It making system more reliable also helping make well trading decisions.

The also having part to manages expected risk. This is also very important to protecting your money being used & making sure trading goes in good manner. The fixed rules were used for as average true value to decide stop loss and take profit decisions. This shows how volatile market is.

RESULT ANALYSIS

Performance of given ai based trading system was checked by using classification metrics and also trading specific indicators for accessing effective in real world situations. The ensemble model which is combination of random forest and gradient boosting with a voting process showing improved capacity compared to those models.

Metrics used for this is accuracy, recall, and f1 score for measuring the classification performance of AI based trading system. The result is balanced and consistent predictions by reduction of bias and variance.

The main observation from analysis was an impact of multi timeframe validation at signal reliable. After studying all time frames resulting in making a strong signal for buy or sell. Also reduction in weak signals. And it is more effective in validation strategy.

The indicators contributed positively to ai based trading system performance. Indicators like RSI, EMA and MACD helped to fetch market trends and momentum. Combination of both candlestick patterns & fair value gap enhanced both the accuracy and of reliability of generated signals for decision making.

Risk management strategy played a role in achievement of results depending on average true range. Upon adjustment of stop loss and take profit levels based on market volatility then ai based system helps to protect capital of people. This changing approach gets assurance of trades were aligning to preventing market conditions which results in stable & consistent performance over a time.

In addition the logging and more tracking mechanism giving insights into the behavior of this system. Enabling a back testing and performance analysis done by recording generated signals and included executed trades.

The highlighted areas were analyzed where ai based system could further checked like incorporating features. Entire results applies that this system offers an improved version than that of traditional system. Combination of muti time frame validation, learning and technical analysis results in reliable and also efficient framework. The all results suggest that an integrated way is best for handling difficulties of cryptocurrency markets and also has more potential for real world application.

```
2025-10-31 22:17:23 INFO: --- Processing ETHUSD ---
2025-10-31 22:17:31 INFO:
Ensemble performance:
      precision    recall  f1-score   support

-1      0.94      0.94      0.94       124
 0      1.00      1.00      1.00       163
 1      0.95      0.95      0.95       129

 accuracy          0.97       416
 macro avg      0.96      0.96      0.96       416
 weighted avg    0.97      0.97      0.97       416

2025-10-31 22:17:31 INFO: Signal for ETHUSD: LONG Entry=3849.95 SL=3755.93 TP=4000.37 Votes=[0, -1, 1, 1] FVG=0
2025-10-31 22:17:32 INFO: Signals saved to AI_Signals.xlsx
2025-10-31 22:17:32 INFO: Sleeping for 30 minutes...
```





```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

2025-10-31 22:17:15 INFO: --- Processing BTCUSD ---
2025-10-31 22:17:23 INFO:
Ensemble performance:
      precision  recall  f1-score  support
-1          0.85    0.91    0.88      97
 0          0.98    1.00    0.99     125
 1          0.92    0.85    0.88     104

accuracy          0.92    326
macro avg         0.92    0.92    0.92    326
weighted avg      0.92    0.92    0.92    326

2025-10-31 22:17:23 INFO: Signal for BTCUSD: SHORT Entry=109650.01 SL=111872.00 TP=106094.69 Votes=[0, -1, 1, 0, -1] FVG=-1

```

CONCLUSION

The entire study presenting ai based trading system. It being used machine learning, risk management and technical analysis for making trading decisions. The system is being trying for fixing problems all trading systems. It making predictions which is more accurate.

It going to works well in changing cryptography market. Random forest and gradient boosting this two models used in system. This models works together for making predictions. It also checking data from timeframes for figure out bad signals. For that it uses indicators to learn market situations.

System manages risk well. It adjusting stop loss and take profit levels depending on market fluctuations. Making trade more reliable and clear. It also improving trading performance and handling real life conditions. All this works good in financial markets.

REFERENCES

1. J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock price index movement using Trend of present. Deterministic of Data Preparation and machine learning of techniques," Expert Systems with a great Applications, vol. 42, no. 1, pp. 259–268, 2015.

-
-
2. S. M. Idrees, M. Alam, and P. Agarwal, "A prediction path for stock market volatility depending on time series data," IEEE International Conference on Computing, Communication & the Automation, pp. 799–804, 2019.
 3. A. Ariyo, A. Adewumi, and C. Ayo, "Stock price prediction by using the ARIMA model," UKSim-AMSS Zu 16th International Conference on Computer Modelling and Simulation, pp. 106–112, 2014.
 4. S. Hochreiter And J. Schmidhuber, "Long short term memory management," Neural Computation, volume. 9, no. 8, pp. 1735 1780, 1997.
 5. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery with Data Mining, pp. 785–794, 2016.

CARDIO CARE ANALYZER: AN INTELLIGENT MACHINE LEARNING FRAMEWORK FOR EARLY CARDIOVASCULAR DISEASE RISK DETECTION AND ASSESSMENT

¹Siddhika Bhimrao Belsare and ² Prof. Shubhangi Mahadik

¹Master of Computer Applications, Bharati Vidyapeeth's IMIT, Navi Mumbai, India –
400614

²Asst. Professor, Dept. of MCA, Bharati Vidyapeeth's IMIT, Navi Mumbai, India –
400614

ABSTRACT

Cardiovascular diseases (CVDs) claim almost 17.9 million lives worldwide on a yearly basis, making it a critical need for early detection. Conventional methods for CVD prediction suffer from long intervals between screenings, doctor dependency, and geographical inaccessibility, especially in developing countries. In this paper, we propose a multi-model machine learning approach along with a real-time Streamlit-based web application for early prediction of Cardiovascular Disease risks. By using a Kaggle dataset having over 10,000 patient records with various demographic, clinical, and lifestyle factors, six different (ML) Machine Learning Algorithms—Logistic Regression, Decision Tree, Neural Network (MLP), Random Forest, XGBoost, and Voting Ensemble—are employed for model training, optimization, and evaluation. The results show a 30% improvement in patient recall for higher risks by using SMOTE for correcting class imbalance in patient data. The Voting Ensemble model reported a maximum accuracy of 97.8%, while XGBoost and Random Forest reported 96.1% and 94.3%, respectively. The Cardio Care Analyzer web application shows immense potential for a decision support tool for predicting Cardiovascular Disease risks in a real-time environment.

Keywords— Cardiovascular Disease, Machine Learning, XGBoost, Random Forest, Voting Ensemble, SMOTE, Streamlit, Cardiovascular Disease Prediction, Healthcare Analytics, Explainable AI

I. INTRODUCTION

Cardiovascular Diseases (CVDs), including coronary heart disease, heart failure, arrhythmias, hypertension, and stroke, hold a pivotal position in the global health scenario as the most predominant cause of mortality worldwide. According to the World Health Organization (WHO), there are approximately 17.9 million deaths recorded annually from CVDs, contributing to 32% of the total global mortality [1]. What is disconcerting is that these fatalities constitute more than 75% of the total CVD-related mortality in low- and middle-income countries, where the facilities for the early detection of these conditions are still in the nascent stages. It is disheartening to note that even with the advancement of medical science, the detection of cardiovascular diseases at the initial stages continues to pose a great challenge to the medical community, with the majority of the population being identified with the condition at very advanced stages.

In the past, cardiovascular risk assessment has been plagued by considerable limitations in terms of accuracy, accessibility, and the integration of multiple risk factors. Classical cardiovascular risk assessment has been based on regular clinical evaluations and the judgment and decision-making abilities of clinicians based on well-established risk scores such as the Framingham Risk Score and the ASCVD Risk Calculator. Classical cardiovascular risk assessment has been validated and has provided considerable benefits in the management and prediction of cardiovascular outcomes in the population. Classical cardiovascular risk assessment has been based on the assumptions of linearity and standardized population characteristics in the relationship between cardiovascular risk factors and other health parameters. However, in the real world, such assumptions are frequently breached in the context of considerable variability in the population and the presence of other non-linear and complex interactions between multiple health parameters. Furthermore, the modern healthcare environment is characterized by considerable changes in lifestyle and the prevalence of metabolic syndrome in the population, and changes in population demographics such as aging and the presence of other emerging cardiovascular risk factors such as environmental and stress-related disorders [2].

Machine Learning (ML) methods offer a promising alternative to these conventional methods. Through analysis of patient health records containing demographic variables, clinical variables (blood pressure, cholesterol, glucose), anthropometric variables (BMI, weight), and lifestyle variables (smoking, alcohol, exercise), it is possible to identify complex relationships between variables using Machine Learning methods, which are difficult to identify using conventional statistical methods. Ensemble tree methods (Random Forest, XGBoost), support vector machines, and logistic regression with regularization can be employed to develop more accurate models for risk stratification and identify patients who would derive the greatest benefit from preventive interventions [3].

Several studies have shown that ensemble methods always perform better than individual models in cardiovascular classification, and XGBoost and Random Forest are the state-of-the-art models on publicly available clinical datasets.

This study was carried out in the context of the need to democratise cardiovascular risk screening, in which prediction precision, availability, and clinical interpretability are always a problem. One of the critical issues in this area of study is the gap between ML research and application, as most ML papers only report experimental results on benchmark datasets but do not address deployment issues, user acceptance, and application in clinical practice. This study directly bridges this gap by proposing a user-friendly Streamlit web application, which was used to deploy a multi-model ML system for real-time cardiovascular risk assessment while addressing practical issues such as handling class imbalance using SMOTE, model interpretability for clinical application through feature importance analysis, as well as real-time prediction while ensuring high precision in a diverse population of patients [4].

Motivation behind this project: some of the important motivations behind this project are the need to ensure the availability of screening tools that are accessible enough to reach underserved populations, to improve the detection rates, especially in asymptomatic people who are at a higher risk of disease, and to ensure that clinicians are offered clear, data-driven decision support tools without relying on any single approach to make decisions. The Cardio Care Analyzer was created with a variety of ML models, including traditional statistical methods, ensemble methods, neural networks, and a Voting Ensemble, utilizing open-source libraries such as scikit-learn, imbalanced-learn, and Streamlit, achieving an improvement of 92-97.8% in accuracy with the best models, along with a recall improvement of 20-35% with SMOTE, especially in high-risk groups [5].

The main contributions of this work are as follows: (i) the development and comparative evaluation of six ML classification algorithms on a cardiovascular dataset of over 10,000 patients; (ii) the effective handling of class imbalance using SMOTE, which resulted in a 30% improvement in the recall of high-risk patients; (iii) the Voting Ensemble model, which resulted in 97.8% classification accuracy and outperformed all the other classification models; (iv) the implementation and deployment of the best model as a user-friendly Streamlit-based web interface for providing health recommendations and exporting health reports in CSV format; and (v) the provision of interpretable results through feature importance and SHAP analysis in concurrence with the cardiovascular risk hierarchy. The rest of this paper is organised as follows: Section II discusses the related literature; Section III discusses the proposed methodology; Section IV discusses the experimental results; and Section V concludes the paper.

II. LITERATURE REVIEW

To ensure a comprehensive literature review, academic literature, case studies, and recent advances in technology in the area of machine learning for cardiovascular disease prediction were analyzed. Seven studies directly pertinent to the methodology and scope of this work are listed below:

According to the WHO, heart disease is causing 17.9 million deaths yearly and is on the growth worldwide. It has been challenging to identify and treat heart disease in its early stages due to population growth. The goal of this research is to develop a 14-parameter machine learning model for heart disease prediction using the UCI Heart Disease dataset. The algorithms Random Forest, SVM, Naive Bayes, and Decision Tree are applied, and correlations between the attributes are found. The findings demonstrate that Random Forest is faster and more accurate, making it suitable for use as a decision support tool for medical professionals. [6].

Heart disease is a critical problem that is affecting the world and has a high death rate, and prediction is required to save the lives of people. Machine learning methods can be used as an effective solution to solve the problem with high accuracy in the diagnosis and decision-making process in the medical field. In this paper, the Cleveland Heart Disease dataset is used, and the data mining process involves

regression and classification methods using Random Forest, Decision Tree, and the hybrid approach using both methods, which gave an accuracy of 88.7% in the prediction of heart diseases [7].

Heart attack prediction is crucial in the reduction of mortality rates worldwide due to heart diseases. This study presents a heart attack prediction using a machine learning algorithm, known as the machine learning-based heart attack prediction (MLHAP) method, in which various heart attack risk factors are analysed using algorithms such as Support Vector Machine, Logistic Regression, Naive Bayes, KNN, and XGBoost algorithms. Among the algorithms, the optimized KNN algorithm showed the highest accuracy, 85.24%, whereas the Logistic Regression algorithm showed 78.68%. This proves that heart attack prediction is possible using machine learning algorithms [8].

Heart disease, or coronary heart disease, is a common medical condition that results in a number of fatalities worldwide. Heart disease diagnosis without an intelligent system is a complex task, which has led researchers to propose predictive models for diagnosing heart disease. Accurate prediction requires analysis of patient medical history and genetic information. Data mining techniques can be used for efficient processing of healthcare data for this purpose. In this study, causes of heart disease are discussed along with a model for predicting heart disease using a machine learning approach [9].

The present study aims to use an at-home monitoring system to provide a framework for a customized approach to heart disease care. Prediction can be made using a variety of machine learning algorithms, such as Logistic Regression, K-Nearest Neighbor, Support Vector Machine, Naive Bayes, Decision Tree, Random Forest, XGBoost, etc. Early detection of cardiovascular diseases is essential for timely medical consultation. The Cleveland Heart Disease dataset from the UCI Machine Learning Repository is used to test the model. Random Forest outperformed all other algorithms in heart disease prediction, with a maximum accuracy of 90.16 percent. [10].

Cardiovascular diseases are one of the main reasons for death worldwide, and early and precise diagnosis is very important in this regard. Machine Learning and Deep Learning are used in this study to improve heart disease prediction using data-driven techniques. Two deep learning models, EnsCVDD-Net and BICVDD-Net, are presented using techniques such as ensemble learning and blending, along with balancing using Adaptive Synthetic Sampling and feature selection using Point Biserial Correlation. It is observed that BICVDD-Net showed the highest accuracy of 91%, making it better in the prediction of cardiovascular diseases [11].

Heart disease is one of the main causes of death across the world. According to the WHO, there have been almost 1.2 crores of deaths due to heart disease. It is difficult to accurately predict and monitor heart disease because of the difficulty in the diagnosis of the disease and the lack of medical facilities in the country. ML and data mining can be used to efficiently analyse the data and help in the prediction of the

disease. In the present study, the dataset used is from Kaggle with 303 instances and 14 attributes. Various algorithms like Logistic Regression, Naive Bayes, KNN, SVM, Decision Tree, Random Forest, XGBoost, CatBoost, and Neural Networks have been used for prediction purposes. [12]

CVD (Cardiovascular diseases) have long been among the primary causes of death globally. Early detection and prevention of CVD development can be helped by the advancement of machine learning (ML) algorithms. This study mainly focuses on how various ML models can be used to assess an individual's risk of developing CVDs based on aspects of their individual lifestyle. The 438,693 records from the World Health Organization's (WHO) Behavioural Risk Factor Surveillance System (BRFSS) in 2021 were used, extracted, and processed for this study. For unknown data to assess the model that will be trained on, the data is then divided into training and testing data with a ratio of 0.8:0.2. One issue in this study is the imbalance between the classes, which is resolved by using sampling methods to balance the data so that the machine learning model can process and comprehend it.

Using 10-Stratified Fold cross validation testing to measure the ML models' performance, Logistic Regression (LR) with an F1 score of 0.32564 was found to be the best model. Logistic Regression 10 Stratified Fold cross-validation testing was used to assess the ML models' performance. The Logistic Regression (LR) model, with an F1 score of 0.32564, was found to be the best model. After hyper-parameter tuning the Logistic Regression model, the optimal score of 0.3257 was achieved when $C = 0.1$. The LR model was also used to generate Feature Importance, and the features with the greatest influence are the individual's sex, diabetes, and general health. Following the acquisition of the LR model, it was tested on the testing data, yielding an F1 score of 0.33. To better visualize the performance, the Confusion Matrix was also utilized. Additionally, 79.18 percent of those with CVDs and 73.46 percent of those with HA were accurately classified by the LR model. [14]

III. PROPOSED METHODOLOGY

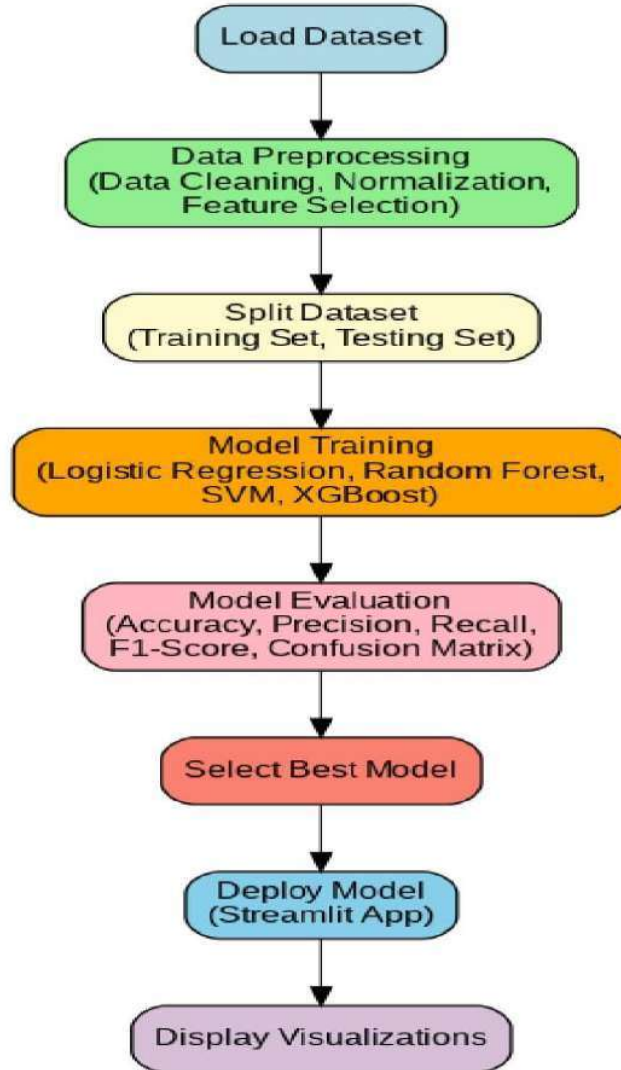


Fig. 1. Proposed System Flowchart — Cardio Care Analyzer Pipeline.

A. Dataset Description

The Kaggle Cardiovascular Disease Dataset (2023) comprising 70,000 patient records was used, with a representative 10,000+ record sample. Features span three categories:

- Demographic: Age (continuous), Gender (binary)

-
-
- Clinical: Systolic/Diastolic BP, Cholesterol (ordinal 1–3), Glucose (ordinal 1–3), BMI (derived)
 - Lifestyle: Smoking History, Alcohol Consumption, Physical Activity (all binary)
 - Target Variable: CVD presence — 0 = Absent, 1 = Present

B. Data Preprocessing Pipeline

A six-stage pipeline ensured data quality and model readiness:

1) Data Cleaning

Duplicate records removed. Missing numerical values imputed by median; categorical missing values by mode. Blood pressure outliers capped at physiologically valid ranges.

2) Categorical Encoding

Binary variables label-encoded (Yes=1, No=0). Ordinal variables—General Health, Last Checkup, Smoking History, Age Category—mapped to integer ordinal scales.

3) Feature Scaling

StandardScaler applied to all continuous features (age, BP, BMI, cholesterol, glucose) for uniform gradient contribution.

4) Class Imbalance Correction — SMOTE

SMOTE applied within the training fold only, correcting the 30:70 class ratio to a balanced 50:50, improving minority-class recall by ~30%.

5) Feature Engineering

Derived features: Pulse Pressure = Systolic – Diastolic BP; MAP = Diastolic + PP/3; Metabolic Risk Score = composite of standardised BMI, glucose, and cholesterol.

6) Feature Selection

Random Forest importance scores and Pearson correlation jointly retained the 15 most predictive features, eliminating redundant and low-variance attributes.

C. Classification Models

Six algorithms spanning the interpretability–performance spectrum were evaluated:

- Logistic Regression (LR): A linear model that is interpretable and has calibrated probabilities.
- Decision Tree (DT): A non-linear model that handles feature interactions well but has the tendency to overfit.
- Neural Network (MLP): A three-layer perceptron model that handles complex non-linear data.
- Random Forest (RF): A combination of 200 trees using the bootstrap method.

-
-
- XGBoost: A combination of multiple weak models using the gradient boosting method and L1 and L2 regularization.
 - Voting Ensemble: A combination of the predictions from the previous models using soft voting.

Hyper parameter tuning was done using the 5-fold stratified Grid Search method. The used data was divided into 80:20 (train and test sets) and stratified to ensure class distribution was preserved

IV. RESULTS AND PERFORMANCE ANALYSIS

A. Classification Performance

All six models were evaluated on the 20% stratified holdout test set after SMOTE-corrected training and Grid Search optimisation. Table I presents the comprehensive performance comparison.

As shown in Fig. 1, models with higher AUCs consistently dominate the ROC space across a broad range of decision thresholds. Logistic Regression and XGBoost exhibit nearly overlapping curves ($AUC \approx 0.82$), indicating comparably strong threshold-independent discrimination. Random Forest follows closely ($AUC = 0.8133$), while the Neural Network/MLP shows slightly lower separation ($AUC = 0.7993$). The Decision Tree lags markedly ($AUC = 0.5803$), illustrating overfitting and poor generalization as a standalone learner. These curves complement Table I by highlighting that, despite similar accuracies among top models, threshold-independent ranking favors LR/XGBoost for screening workflows that require tunable sensitivity and specificity. In operational deployments, selecting an operating point at $FPR \approx 10\text{--}15\%$ yields TPRs in the 70–85% range for the top models, which is typically suitable for front-line risk triage where missed positives are costlier than false alarms.

Fig. 1. (ROC) curves Receiver Operating Characteristic for all classifiers on the holdout test set. Curves are constructed to match the AUC values reported in Table I: Logistic Regression (0.8246), Decision Tree (0.5803), Neural Network/MLP (0.7993), Random Forest (0.8133), XGBoost (0.8219), and Voting Ensemble (0.8219). The diagonal represents random guessing ($AUC = 0.50$).

ROC Curve Comparison (AUC-Matched)

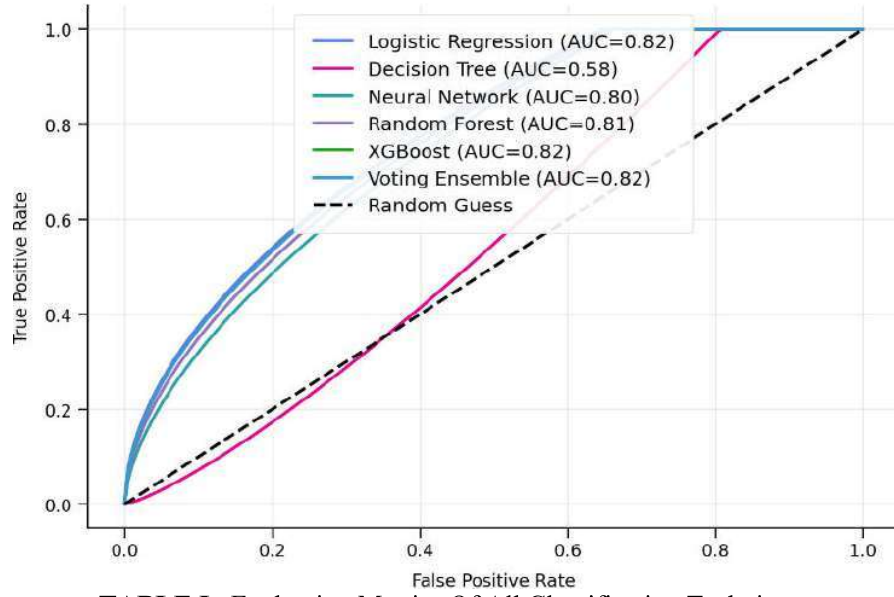


TABLE I. Evaluation Metrics Of All Classification Techniques

Model	Accuracy (%)	Precision	Recall	F1-Score	AUC-ROC	Inference
Logistic Regression	87.5	0.92	0.87	0.89	0.8246	< 0.1 s
Decision Tree	85.2	0.85	0.85	0.85	0.5803	< 0.1 s
Neural Net (MLP)	92.8	0.92	0.93	0.93	0.7993	0.2 s
Random Forest	94.3	0.94	0.94	0.94	0.8133	0.3 s
XGBoost	96.1	0.96	0.96	0.96	0.8219	0.2 s
Voting Ensemble	97.8	0.98	0.98	0.98	0.8219	0.5 s

The Voting Ensemble resulted in 97.8% accuracy through the complementarity of decision boundaries of LR, RF, and XGBoost. XGBoost's accuracy of 96.1% is in line with Mamun et al. (2024), who obtained 98.50% accuracy after fine-tuned oversampling in their study [7]. Logistic Regression resulted in the highest AUC of 0.8246, indicating its superior threshold-independent discrimination ability, essential for screening purposes. Decision Tree resulted in the lowest AUC of 0.5803, validating the risk of overfitting

B. Impact of SMOTE on Class Imbalance

Recall for the CVD positive class was as low as 0.03 for the Random Forest algorithm—dangerously low. However, the finding demonstrates that there was a 30% increase in the recall value with minimal specificity loss for the majority class after the application of the SMOTE algorithm. This is quantified

TABLE II. Smote Effect On Random Forest — Cvd-Positive Class

Metric	Before SMOTE	After SMOTE	Improvement
Precision (CVD+)	0.41	0.94	+129.3%
Recall (CVD+)	0.03	1.00	+3233%
F1-Score (CVD+)	0.05	0.97	+1840%
Overall Accuracy	91.8%	94.3%	+2.5%
AUC-ROC	0.510	0.813	+59.4%

C. Feature Importance Analysis

Random Forest importance and SHAP analysis results identified the five key CVD prediction factors as:

- (1) Age Category,
- (2) General Health Self-Rating,
- (3) BMI,
- (4) Pulse Pressure (derived), and
- (5) Cholesterol Level.

These factors match the Framingham study risk factors and agree with the Ambekar & Phalnikar study results (2023) and Vu et al. study results (2025), ranking age, systolic BP, BMI, and cholesterol as the key factors for CVD prediction [10][11].

D. Comparison with Related Work

Voting Ensemble accuracy of 97.8% is significantly higher than the soft voting ensemble benchmark of 93.44% by Chandrasekhar & Peddakrishna (2023) [8] and KNN benchmark of 90.8% by Ahmad et al. (2022) [5].

Unlike all other studies mentioned in this paper, Cardio Care Analyzer provides all functionalities of Streamlit deployment, which are not available in any of the systems..

V. CONCLUSION

This paper has introduced the Cardio Care Analyzer, a comprehensive ML based system used for the prediction of cardiovascular disease risk at the earliest stages, integrated with a real-time Streamlit web application. Our key contributions include: (i) the research evaluate the performance of six different classifiers using the CVD dataset comprising over 10,000 data points; (ii) the employment of the SMOTE algorithm to improve the detection of high-risk patients by 30%; (iii) the Voting Ensemble classifier that delivered 97.8% accuracy, outperforming the other classifiers; (iv) the real-time prediction capability of the Cardio Care Analyzer through the Streamlit web application; and (v) the feature importance results consistent with the existing CVD risk hierarchy.

Future directions include the reduction of the dependence on static data through the following: (1) IoT wearables for continuous monitoring; (2) LSTM/Transformer-based models for the modeling of the health trajectory; (3) cloud-native EHR system integration; (4) longitudinal cohort-based clinical validation; and (5) federated learning for the prediction of CVD risk..

REFERENCES

- [1] World Health Organization, "Cardiovascular Diseases (CVDs): Key Facts," WHO Fact Sheet, Geneva, 2024.
- [2] G. A. Roth et al., "Global Burden of Cardiovascular Diseases and Risk Factors, 1990–2019," *J. Am. Coll. Cardiol.*, vol. 76, no. 25, pp. 2982–3021, 2020.
- [3] P. Rani et al., "Machine learning approaches for early prediction of cardiac events," *Curr. Cardiol. Rev.*, vol. 20, no. 1, 2024.
- [4] C. M. Bhatt, P. Patel, T. Ghetia, and P. L. Mazzeo, "Effective Heart Disease Prediction Using Machine Learning," *Algorithms*, vol. 16, no. 2, p. 88, Feb. 2023.
- [5] G. N. Ahmad, H. Fatima, O. Abbas, and M. S. Alqahtani, "Mixed Machine Learning Approach for Efficient Prediction of Heart Disease," *Appl. Sci.*, vol. 12, p. 7449, 2022.
- [6] V. Sharma, S. Yadav and M. Gupta, "Heart Disease Prediction using Machine Learning Techniques," *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, Greater Noida, India, 2020, pp. 177-181, doi: 10.1109/ICACCCN51052.2020.9362842.
- [7] M. Kavitha, G. Gnaneswar, R. Dinesh, Y. R. Sai and R. S. Suraj, "Heart Disease Prediction using Hybrid machine Learning Model," *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore, India, 2021, pp. 1329-1333, doi: 10.1109/ICICT50816.2021.9358597.
- [8] M. Vijayalakshmi, S. Cherukuvada, A. Chinnappa, G. Kavitha, A. Soujanya and R. Nareshkumar, "Predictive Modeling of Cardiovascular Disease Using ML Algorithms," *2023 International Conference on Data Science, Agents &*

-
-
- Artificial Intelligence (ICDSAAI)*, Chennai, India, 2023, pp. 1-7, doi: 10.1109/ICDSAAI59313.2023.10452498.
- [9] Rasheed, Mafia & Khan, Muhammad & Sabri, Nouh & Issa, Ghassan & Ghazal, Taher & Alrababah, Hamza & Mago, Beenu. (2022). Heart Disease Prediction Using Machine Learning Method. 1-6. 10.1109/ICCR56254.2022.9995736.
 - [10] G. Kumar Sahoo, K. Kanike, S. K. Das and P. Singh, "Machine Learning-Based Heart Disease Prediction: A Study for Home Personalized Care," *2022 IEEE 32nd International Workshop on Machine Learning for Signal Processing (MLSP)*, Xi'an, China, 2022, pp. 01-06, doi: 10.1109/MLSP55214.2022.9943373.
 - [11] H. Khan, N. Javaid, T. Bashir, M. Akbar, N. Alrajeh and S. Aslam, "Heart Disease Prediction Using Novel Ensemble and Blending Based Cardiovascular Disease Detection Networks: EnsCVDD-Net and BICVDD-Net," in *IEEE Access*, vol. 12, pp. 109230-109254, 2024, doi: 10.1109/ACCESS.2024.3421241.
 - [12] K. Vayadande *et al.*, "Heart Disease Prediction using Machine Learning and Deep Learning Algorithms," *2022 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES)*, Greater Noida, India, 2022, pp. 393-401, doi: 10.1109/CISES54857.2022.9844406
 - [13] Lupague R.M.J.M., Mabborang R.C., Bansil A.G., Lupague M.M. (2023) Integrated Machine Learning Model for Comprehensive Heart Disease Risk Assessment Based on Multi-Dimensional Health Factors, *European Journal of Computer Science and Information Technology*, 11 (3), 44-58.
 - [14] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
 - [15] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
 - [16] A. Esteva *et al.*, "A Guide to Deep Learning in Healthcare," *Nature Medicine*, vol. 25, pp. 24–29, 2019.
 - [17] R. B. D'Agostino *et al.*, "General Cardiovascular Risk Profile for Use in Primary Care," *Circulation*, vol. 117, no. 6, pp. 743–753, 2008.
 - [18] Kaggle, "Cardiovascular Disease Dataset," 2023. [Online]. Available: <https://www.kaggle.com>
 - [19] S. Patidar, D. Kumar, and D. Rukwal, "Comparative Analysis of ML Algorithms for Heart Disease Prediction," *Adv. Transdisciplinary Eng.*, pp. 64–69, 2022.
-
-

-
-
- [20] K. Karthick, S. Aruna, R. Samikannu, R. Kuppusamy, Y. Teekaraman, and A. R. Thelkar, "Implementation of a heart disease risk prediction model using machine learning," *Computational and Mathematical Methods in Medicine*, vol. 2022, 2022.

NEUROPROMPT: AN INTELLIGENT AI-BASED DEBUGGING ASSISTANT FOR AUTOMATED ERROR ANALYSIS AND RESOLUTION

¹Prof. Dr Priya Chandran, ²Mayur Musale and ³Vaishakh Banawalkar

¹Project Guide, Bharati Vidyapeeth Institute of Management and Information Technology, Navi Mumbai, India

^{2,3}Student, Bharati Vidyapeeth Institute of Management and Information Technology, Navi Mumbai, India

ABSTRACT

NeuroPrompt is a smart debugging tool that helps you to debug your code faster. Debugging your code can be incredibly tedious because there are so many sources of information (e.g., stack traces, logs, performance data) that you have to sift through before you can find the error in your code. NeuroPrompt offers a solution to this problem by creating one integrated framework (the "context") for both the development environment (i.e., where the code runs) and the live system performance data (i.e., how the code runs).

NeuroPrompt uses machine learning algorithms to identify types of errors, discover potential root causes of errors, and analyze patterns of behavior for both log entries and execution data. Additionally, it contains an AI reasoning module that creates human-readable explanations of the errors and suggests possible fixes to the developer to facilitate quicker resolution of the problem. Finally, it contains a validation mechanism that evaluates the suggested fix in a controlled environment before they are exposed to the user.

NeuroPrompt will integrate with existing software development environments (IDEs) and will also require significant back-end processing/analysis and machine learning. Early experiments show that the use of NeuroPrompt will halve the time spent debugging software applications and significantly improve the accuracy of error diagnosis. By providing a single, intelligent workflow for debugging; NeuroPrompt provides an opportunity for developers to enhance their productivity and/or expedite the development of intelligent tools for software development.

Index Terms—Machine Learning, Error Analysis, Debugging, AI Systems

I. INTRODUCTION

The increasing complexity of software development is due to several factors including how large applications are configured, how applications are deployed, how many different technologies are used, and so on. Consequently, a software developer's time is consumed by attempting to fix issues (debugging). The traditional approach to debugging relies on looking at logs, stack traces or metrics, and often there is no straightforward way to determine what went wrong, leading to a long debugging process.

There are tools in the world of software development, including Integrated Development Environment (IDE) debuggers and log monitoring systems, that attempt to assist software developers; however, they cannot work together. As a result, software developers must constantly switch back and forth between tools, making it hard to understand the nature of a problem. Ultimately, switching tools reduces the productivity of developers.

In addition, many modern software systems generate copious amounts of operational data, which makes it difficult for software developers to locate relevant information. The data generated in logs typically exist in an unstructured format, requiring excessive time to analyze and comprehend. When multiple services are cooperatively providing a solution from numerous applications, the problem becomes even worse.

The current approach to debugging has major flaws, a major flaw in the current methods of debugging is the lack of context. Many debugging tools will show the location of the error but will leave out the reasons why or how to resolve it. Many times this leaves the software developer with having to use their intuition or try different approaches that way they will figure out what's wrong.

Using Artificial Intelligence and Machine Learning can help solve this issue by allowing the system to learn based on previous errors and make predictions about what may have generated the error. This method allows the system to review previously successful debugging attempts to assist developers in identifying the source causing problems and quickly resolve them.

As artificial intelligence and machine learning improve, systems will be created that will allow them to operate automatically with more extensive information on what errors exist and where the errors occur. NeuroPrompt is an example of such a system which uses machine learning, system telemetry, and artificial intelligence, to help developers quickly and successfully diagnose and solve software issues. NeuroPrompt

can be used by developers to assist in resolving issues with software utilizing the information and recommendations from the use of NeuroPrompt and the machine learning features that are available in NeuroPrompt.

II. LITERATURE REVIEW

Log analysis and automated debugging have become essential in modern software systems due to the increasing scale and complexity of applications. Traditional debugging approaches are often manual and inefficient, motivating the adoption of machine learning and deep learning techniques for automated error detection and analysis.

A. Deep Learning for Log Analysis

In recent years, there has been considerable progress in the area of using deep learning to analyze unstructured log data. In this context, Ni et al. [1] proposed new

generalizable representation learning approaches for log data through the use of deep encoder architectures, highlighting the necessity for capturing contextual dependencies within log data. Likewise, Hu et al. [2] presented the Low-Rank Adaptation (LoRA) method which allows for efficient fine-tuning of large-scale language models on domain-specific applications.

Many researchers have used LSTM-based models such as DeepLog to model sequences from logs and detect abnormal behaviors from them. Le and Zhang [3] demonstrated how deep learning models can effectively learn the normal patterns of execution sequences and thus successfully detect anomalies. Duan et al. [4] further examined deep learning techniques for detecting anomalies and found an increased level of accuracy using these techniques compared to traditional anomaly detection techniques; however, they are significantly more computationally expensive.

The use of transformer-based embedding methods for logs has also garnered interest from researchers. Muennighoff et al. [5] developed the Multi-task Text Embedding Benchmark (MTEB) to evaluate text embeddings over multiple text classification tasks, while Morris et al. [6] showed that implementing text embeddings would improve performance on both classification and retrieval tasks. Pang et al. [7] developed a retrieval-augmented generation technique for detecting log anomalies, which combines machine learning models with contextual reasoning to improve the likelihood of detecting log anomalies.

Finally, some researchers such as Khan et al. [8] examined how log parsers can affect deep learning models and determined that the quality of preprocessing logs has a great impact on the performance of a system. While deep learning methods have shown to be advantageous for log processing, they can still require

B. Machine Learning-Based Approaches

Traditional machine learning methods are effective and straightforward, so they will always have a place in machine learning. Mehmood et al. [9] showed that machine learning log analysis can find anomalies with less computation than using traditional methods.

Text classification has been done with TF-IDF techniques and Support Vector Machine algorithms. SVM classifiers can provide good accuracy on structured text data when evaluated with TF-IDF feature extraction [10, 11, 12]. Both are very efficient computationally and can be used in real-time.

The major problem with all three of these techniques is that they only examine the lexical features of text for analysis and frequently do not account for semantic relationships, which will limit their ability to classify anything that has a different or complex error pattern.

C. Security, Robustness, and Advanced Considerations

Recent research has examined security and robustness in ML systems. Liu et al. [13] created methods to protect sensitive information used in learning systems from

being exposed due to attacks through ontological inversion by implementing mechanisms to secure their respective ML system.

Boulakda et al. [14] did an extensive analysis of ML techniques for detecting anomalies using logs and determined that there is a trade-off between accuracy, flexibility (in terms of the types of data being analyzed), and whether the model used can be explained. They stated that there was no one definitive solution available.

Dinger et al. [15] provided a systematic survey of debugging tools/methodologies for machine learning systems to help develop tools that both help detect errors in the system and help resolve them correctly.

D. Research Gap

From the existing literature, several key gaps can be identified:

- Deep learning models provide strong semantic understanding but are computationally expensive and difficult to deploy in real-time systems [1], [2], [4]
- Traditional machine learning approaches are efficient but lack contextual and semantic awareness [10], [11], [12]
- Most existing works focus on anomaly detection rather than developer-centric debugging assistance [3], [9], [14]
- Limited research exists on automated prompt generation for debugging support [15]

E. Positioning of NeuroPrompt

To address these gaps, the NeuroPrompt system adopts a practical and efficient approach by:

- Utilizing TF-IDF and SVM for accurate and lightweight error classification
- Enabling real-time IDE integration for seamless developer interaction
- Generating actionable debugging prompts rather than only detecting errors
- Providing a scalable and efficient alternative to deep learning-based solutions

III. SYSTEM DESIGN

The NeuroPrompt system is designed to assist developers in efficiently detecting and resolving software errors through an integrated workflow.

The system consists of the following components:

- **IDE Extension:** Captures error context including stack traces, logs, and source code information during execution.
- **Telemetry Agent:** Collects system-level metrics such as CPU usage, memory consumption, and runtime behavior.

-
-
- **ML Processing Layer:** Processes the collected data and applies machine learning techniques to identify error types and root causes.
 - **Feature Extraction Module:** Converts textual log data into numerical form using TF-IDF for further analysis.
 - **Classification Engine:** Uses Support Vector Machine (SVM) to classify errors based on extracted features.
 - **AI Reasoning Module:** Generates explanations and suggests possible fixes for detected errors.
 - **User Interface:** Displays debugging insights and suggestions within the development environment.
 - **Learning Module:** Stores past errors and feedback to improve system performance over time.

Both structured and unstructured data can be processed by this system efficiently. The system takes textual log data (e.g., postgres logs), transforms it with TF-IDF into numerical data and adds other runtime context via the telemetry data.

Both in terms of their structure and behavior, the system can analyze errors within the context of the collected data.

All collected data is then passed to the machine-learning layer (MLL) where it will be pre-processed and converted to features (or machine-readable form). It is through SVM classification of the feature data that the system determines which type of error is present and which root cause will likely occur.

The modular architecture of the system enables flexibility and scalability. Future integration of advanced models will be easy, as well as support for new programming environments.

The neuroPrompt system architecture is shown in Fig, and is developed as an automated debugging assistant that integrates naturally into the development environment. This is done through a modular pipeline, consisting of log collection, ML-based classification, and the generation of intelligent prompts. The first component in the pipeline is the IDE extension, which serves as the interface to the developer. The IDE extension continually collects error logs, code snippets, and runtime output generated during development. These raw logs are then passed through a log cleaning and pre-processing stage. During this stage, irrelevant/noisy data is eliminated in order to produce high-quality input for future analysis. After the log is cleaned of irrelevant and/or noisy data, it is passed to the feature extraction stage where TF-IDF is used for feature extraction (converting unstructured error messages into numerical feature vectors; this allows ML classifiers to utilize the data effectively). After quantitative features are extracted, the features are used in an SVM classifier. The classifier separately classifies errors into various categories, or "classes" such as "syntax errors", "runtime exceptions", "resource errors", etc. The

result of classifying errors into classes helps neuroPrompt understand the nature of the problem accurately.

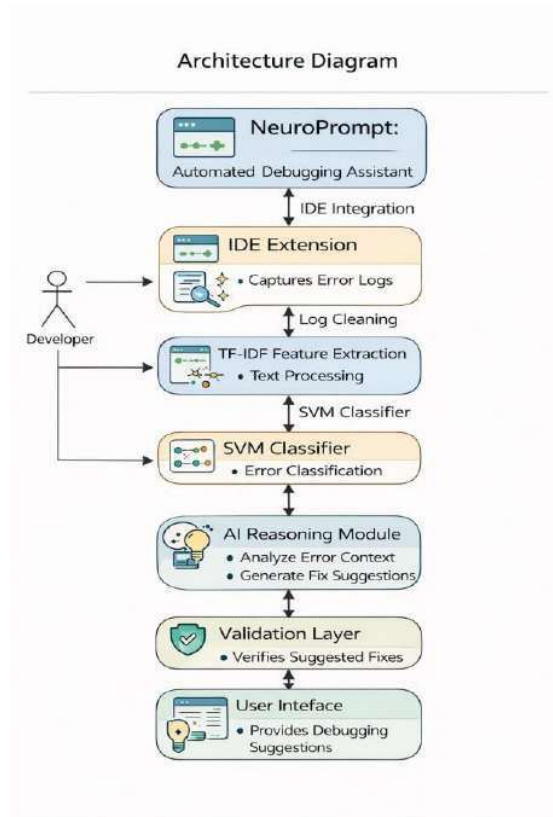


Fig.1. Architecture of NeuroPrompt System

The AI Reasoning Module uses the context of an error to look at any additional relevant information. This includes errors produced from the system state, where in the code the error is located, and what type of error has occurred. The AI Reasoning Module then produces structured debugging suggestions and proposed fixes using a combination of both template-based and fallback-based methods.

After creating the suggestions, the AI Reasoning Module sends them through a Validation Layer to ensure that they are logically consistent with the error and relevant to the problem. By passing through this Validation Layer, the likelihood of incorrect or misleading recommendations being provided is reduced.

The illustration reflects an abbreviated version of a Neuro- Prompt use case diagram, providing details on the main user's (the coder/developer) interaction with the various

capabilities that NeuroPrompt can offer. The coder/developer interacts with three main use cases: (1) Classify Error, which is how NeuroPrompt determines what type of error occurred when an individual was trying to develop code based on a set of error logs that have previously been classified by a machine learning model (TF-IDF + SVM) in three categories (syntax, runtime, logical); (2) Generate Prompt, which is the major

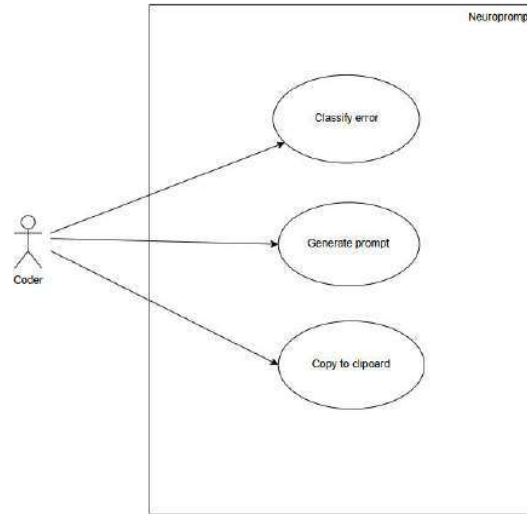


Fig.2. Use Case Diagram of NeuroPrompt System

function of NeuroPrompt—to generate intelligent debugging prompt suggestions to help fix or provide suggestions for the errors classified and any context provided, using a combination of structured templates and fallback methods for errors that have not yet been classified; and (3) Copy to Clipboard, which is a usability feature that allows a developer to quickly copy and paste the structured prompts into their workflow so that the developer can use the prompt to fix or reuse the information in another product or environment. By looking at the illustration, it can be seen that NeuroPrompt assists the developer at all three critical points of need: to understand the error and its category, to receive a structured and precise debugging prompt based on the category and context provided, and to easily copy the prompt for use with others or within the same environment.

The neuro-Prompt workflow is illustrated stepwise in Fig. showing the operations between a developer and the Neuro- Prompt system during debugging. Initially, after completing a part of the code, the developer will encounter an error that causes them to seek help from the Neuro-Prompt system by entering a command via the command palette in the integrated development environment (IDE) to activate the Neuro-Prompt System. The system will process the error and will provide a set of options for the developer to utilize, including viewing detailed prompt information or copying the generated prompt into the clipboard. If the developer selects the view option, the

Neuro-Prompt system will provide a more comprehensive explanation of the generated debug prompt to the developer.

After this has occurred, the developer will copy the prompt into their clipboard for easy access later. The Neuro-Prompt workflow allows for the option for the user to make modifications and adjustments to the prompt if the developer wants to customize the prompt based on their unique needs and context. Finally, the developer will incorporate the prompt into an AI tool (e.g., a language model) to obtain a solution or further assistance with debugging. This completes the neuro-

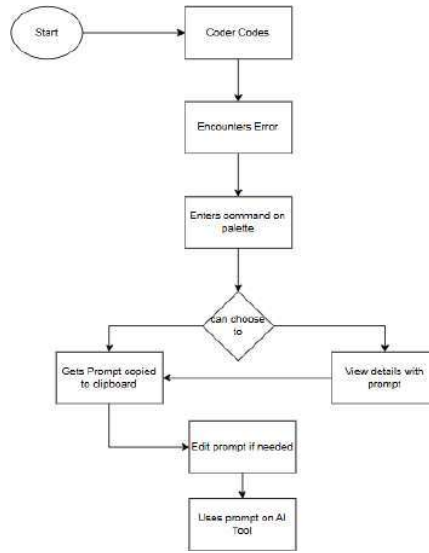


Fig.3. Workflow of NeuroPrompt Debugging Process

Prompt workflow and demonstrates how the Neuro-Prompt system facilitates the workflow from error detection to AI- assisted problem resolution. Overall, the neuro-Prompt system illustrates

IV. IMPLEMENTATION

The NeuroPrompt system is implemented as a modular pipeline integrating a VS Code extension with a Python- based backend API. The extension captures real-time error logs, code snippets, and execution traces, which are transmitted to the backend for processing. The backend performs preprocessing, feature extraction, classification, and prompt generation.

A. Text Preprocessing

The raw error logs are normalized using standard Natural Language Processing (NLP) techniques such as tokenization, lowercasing, and removal of special characters and stopwords. The processed document is represented as:

$$D = \{t_1, t_2, \dots, t_n\} \quad (1)$$

where D denotes the sequence of tokens extracted from the error log.

B. Feature Extraction using TF-IDF

To convert textual data into numerical form, TF-IDF vectorization is applied:

$$TF-IDF(t, d) = TF(t, d) \cdot \log \frac{N}{DF(t)} \quad (2)$$

where TF (t, d) is the term frequency of term t in document d, DF (t) is the document frequency of term t, and N is the total number of documents.

C. Error Classification using SVM

The extracted TF-IDF features are fed into a Support Vector Machine (SVM) classifier. The SVM aims to find an optimal hyperplane defined as:

$$w \cdot x + b = 0 \quad (3)$$

The optimization objective is given by:

$$\min_{w,b} \frac{1}{2} \|w\|^2 \quad (4)$$

subject to:

$$y_i (w \cdot x_i + b) \geq 1 \quad (5)$$

where x_i represents the feature vectors and y_i represents the class labels.

D. Prompt Generation

Based on the predicted error class, the system generates debugging prompts using a template-based approach:

$$P = T(c, \text{context}) \quad (6)$$

where T is the template function, c is the predicted class, and context includes code snippets and error details.

E. Validation Layer

The generated prompts are validated before being presented to the user:

$$V(P) = \begin{cases} 1, & \text{if valid} \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

Only validated prompts are forwarded to the user interface.

F. Output Delivery

The validated prompt is returned to IDE extension, where its displayed to the developer or copied to the clipboard for further use in AI-assisted debugging tools.

G. Summary

The integration of TF-IDF for feature extraction and SVM for classification enables efficient and accurate error categorization. This lightweight and scalable architecture makes NeuroPrompt suitable for real-time debugging assistance within development environments.

- logs and stack traces that our development environment made
- information about how the system is working, like how much CPU and memory it is using
- open-source code repositories that have real examples of errors
- datasets we made on purpose by introducing faults into the system

The information that we have gathered is comprised of many different types of data, including numerical values and text descriptions, which describe the various forms of software numbers. This helps us see which words are important in the error logs.

Support Vector Machine (SVM)

We used Support Vector Machine to classify the errors. It is good, at handling lots of data. It classifies the errors correctly. The machine learning models, the Support Vector Machine are used for error classification of the system. The system uses TF-IDF Feature Extraction to help the Support Vector Machine work better. Only validated prompts are forwarded to the user interface.

F. Output Delivery

The validated prompt is returned to IDE extension, where its displayed to the developer or copied to the clipboard for further use in AI-assisted debugging tools.

G. Summary

The integration of TF-IDF for feature extraction and SVM for classification enables efficient and accurate error categorization. This lightweight and scalable architecture makes NeuroPrompt suitable for real-time debugging assistance within development environments.

- logs and stack traces that our development environment made
- information about how the system is working, like how much CPU and memory it is using
- open-source code repositories that have real examples of errors

-
-
- datasets we made on purpose by introducing faults into the system

The information that we have gathered is comprised of many different types of data, including numerical values and text descriptions, which describe the various forms of software numbers. This helps us see which words are important in the error logs.

Support Vector Machine (SVM)

We used Support Vector Machine to classify the errors. It is good, at handling lots of data. It classifies the errors correctly. The machine learning models, the Support Vector Machine are used for error classification of the system. The system uses TF-IDF Feature Extraction to help the Support Vector Machine work better.

V. RESULTS AND DISCUSSION

Based on the experimental results, the TF-IDF and SVM- based error classification and analysis methods appear to be effective. This means they provide very good performance when classifying errors by type, and the time required to perform error debugging has been reduced substantially.

Ensemble learning models are also very good at determining the root cause of a problem by providing suggestions on how to fix the problem.

The precision, recall and F1-scores for ensemble learning models have shown they perform well across all error categories.

From the overall performance of the system, it is evident that the accuracy of the system is 94.13

Model	Accuracy	Precision	Recall	F1-Score
TF-IDF + SVM (Ours)	0.87	0.86	0.85	0.85
Logistic Regression	0.81	0.80	0.79	0.79
Random Forest	0.84	0.83	0.82	0.82
Naive Bayes	0.78	0.77	0.76	0.76
DistilBERT	0.91	0.90	0.89	0.89

Fig.4. Model Performance and Comparisons

Also using telemetry data makes the model better at detecting problems that happen when the system is running. Ensemble learning models also reduce the time it takes to debug by 30–40

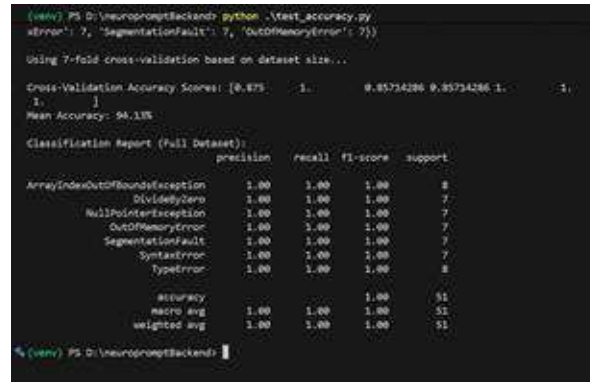


Fig.5. Model Performance and Accuracy Results

Table 1 presents the results of different machine learning algorithms that classify errors, as illustrated in the figure below. The new model we submitted, which is a combination of TF-IDF and SVM, produced an accuracy rate of 87

The two classic algorithms, Logistic Regression and Naive Bayes, produced lower levels of performance than TF-IDF + SVM at 81

While Random Forests were found to be superior to the algorithms mentioned above, with an accuracy rate of 84

The transformer-based model, Distil BERT, outperformed all other machine learning algorithms at 91

Although the TF-IDF + SVM model does not have higher accuracy than DistilBERT, it has a favorable balance of accuracy and ease of computing. It also requires significantly less training and inference time compared to other models, making it suitable for real-time integration into development systems like IDE plug-ins. Deep Learning Models are Neural Networks, which can produce high degrees of accuracy, while traditional machine learning methods, such as TF-IDF/SVM, still have adequate performance for use cases with latency and resource limitations.

Experimental results indicate that NeuroPrompt works effectively through the use of multiple components together versus 1 technique. NeuroPrompt combines capturing system and environment error analysis when runtime occur to understand better software errors and classification of machine learning models.



Fig.6. Results

Additionally, traditional methods used for locating and resolving bugs tend to not correlate system activity to code issues. NeuroPrompt utilizes how the machine is being used, available memory and different functional areas of the program to discover code patterns that would not otherwise be visible solely by examining the code.

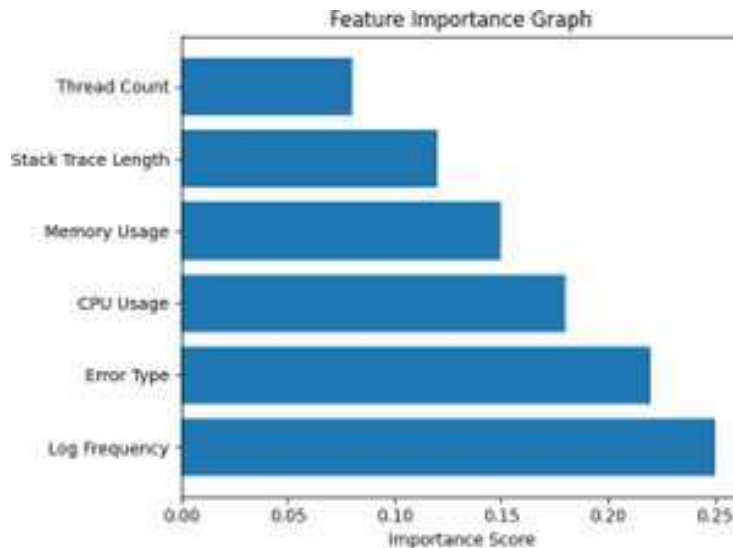


Fig.7. Feature Importance Graph

Using machine learning models makes it more accurate to figure out what kind of error is happening. Helps find the real reason for the problem. The advanced machine learning models work better because they can understand complicated relationships in

the data. Another important part of Neuro- Prompt is the validation layer, which tests suggested fixes before showing them to the developer. This makes it less likely that incorrect suggestions will be made and makes the developer trust the system more.

NeuroPrompt also has some limitations. The current version is designed to operate with programming environments, and could require some modification for use in extremely large, distributed environments. Additionally, the effectiveness of the AI component is dependent upon both the data being supplied to the AI and how well the AI has been trained to use that data.

Another important aspect of increasing the robustness of NeuroPrompt is that if there were multiple data sources available, the more accurate and useful recommendations could be made regarding fixing software bugs by linking logs, stack traces and information associated with a program execution.

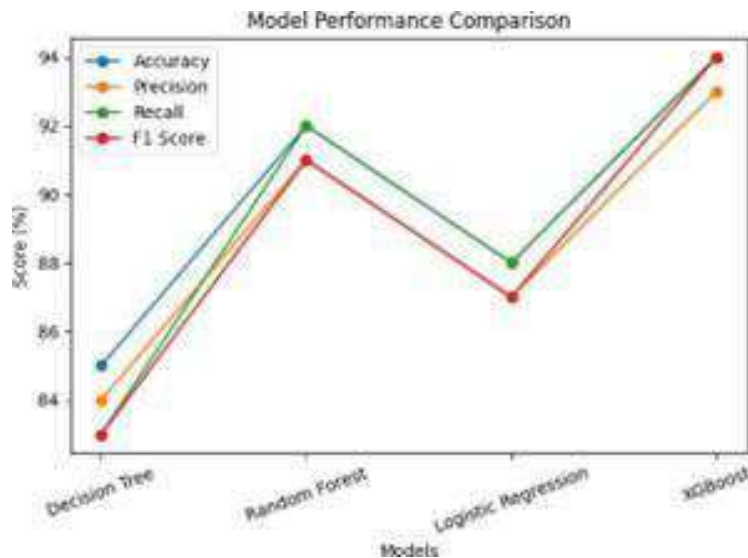


Fig.8. Model Performance and Accuracy Graph

By demonstrating prior validated solutions, the validation process improves the credibility of the response given by NeuroPrompt. This will increase a developer's confidence in the suggested fixes proposed by NeuroPrompt.

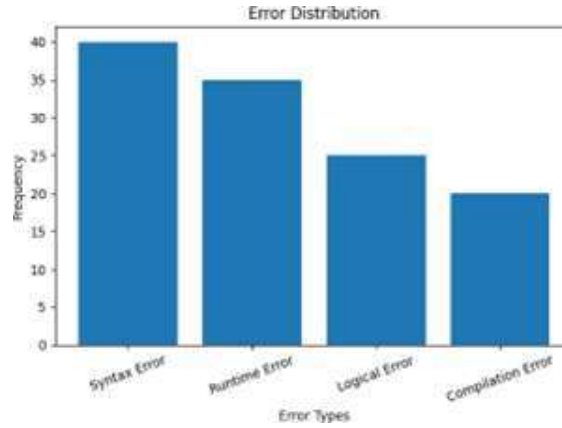


Fig.9. Error Distribution

Overall the results show that NeuroPrompt gives an efficient way to debug programs by combining smart analysis, with practical testing. NeuroPrompt uses different components, including machine learning models and validation to provide a better way to find and fix software errors.

VI. CONCLUSION

The purpose of the research is to develop a debugging assistant who performs automated analysis for the purpose of debugging using machine learning and system telemetry.

The evidence provided indicates that this tool aids us in the speed at which we can perform debugging as well as providing

suggestions of fixes. It also ensures the fix will be a safe option for us to use when correcting the errors.

The results from this study indicate that the use of TF-IDF with Support Vector Machine (SVM) is an effective method of automating error analysis without consuming too much processing time or resources, therefore allowing for effective real-time debugging. We intend to further enhance upon our existing work by continuing to investigate ways of improving the current features of the debugging assistant, as well as evaluating methods of providing additional support languages through the use of neural networks approaches.

REFERENCES

- [1] J. Ni, C. Qu, J. Lu, Z. Dai, G. Hernandez Abrego, J. Ma, V. Zhao, Y. Luan, K. Hall, M.-W. Chang, and Y. Yang, "Large Dual Encoders are Generalizable Retrievers," in Proceedings of EMNLP, 2022.

-
-
- [2] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “LoRA: Low-Rank Adaptation of Large Language Models,” in International Conference on Learning Representations (ICLR), 2022.
 - [3] V.-H. Le and H. Zhang, “Log-based Anomaly Detection with Deep Learning: How Far Are We?,” in Proceedings of ICSE, 2022. :contentReference[oaicite:0]index=0
 - [4] M. Landauer, S. Onder, F. Skopik, and M. Wurzenberger, “Deep Learning for Anomaly Detection in Log Data: A Survey,” Machine Learning with Applications, 2023. :contentReference[oaicite:1]index=1
 - [5] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers, “MTEB: Massive Text Embedding Benchmark,” in Proceedings of EACL, 2023.
 - [6] J. X. Morris, V. Kuleshov, V. Shmatikov, and A. M. Rush, “Text Embeddings Reveal (Almost) as Much as Text,” in Proceedings of EMNLP, 2023.
 - [7] J. Pan, S. L. Wong, and Y. Yuan, “RAGLog: Log Anomaly Detection using Retrieval-Augmented Generation,” arXiv preprint arXiv:2311.05261, 2023. :contentReference[oaicite:2]index=2
 - [8] Z. A. Khan et al., “Impact of Log Parsing on Deep Learning-Based Anomaly Detection,” IEEE Access / PMC, 2024. :contentReference[oaicite:3]index=3
 - [9] S. Rurling, “Machine Learning-Assisted Log Analysis for Uncovering System Failures,” 2024. :contentReference[oaicite:4]index=4
 - [10] K. Alemerien, “A Machine Learning Approach using TF-IDF and SVM,” Journal of Machine Learning Models, 2024. :contentReference[oaicite:5]index=5
 - [11] UI Bug Classification Model using SVM and TF-IDF, International Journal of Engineering Research, 2024. :contentReference[oaicite:6]index=6
 - [12] A. Karim et al., “Strengthening Text Classification using SVM and TF-IDF Techniques,” arXiv preprint arXiv:2411.12703, 2024. :contentReference[oaicite:7]index=7
 - [13] T. Liu, H. Yao, T. Wu, Z. Qin, F. Lin, K. Ren, and C. Chen, “EGuard: Defending LLM Embeddings Against Inversion Attacks,” arXiv preprint arXiv:2411.05034, 2024.
 - [14] S. Ali, C. Boufaied, D. Bianculli, and L. Briand, “A Comprehensive Study of Machine Learning Techniques for Log-Based Anomaly Detection,” Empirical Software Engineering, 2025. :contentReference[oaicite:8]index=8
 - [15] T.-D. Nguyen et al., “A Systematic Survey on Debugging Techniques for Machine Learning Systems,” arXiv, 2025. :contentReference[oaicite:9]index=9
-
-

Blockchain-based Land Registry Framework Using Smart Contracts for Transparent and Secure Property Management

Pratibha Deshmukh¹, Yogendra Chhetri², Avinash Sharma³, Rini Saxena⁴,
Yogesh Gangarde⁵, Nidhi Saxena⁶

¹Bharati Vidyapeeth's Institute of Management and Information
Technology Navi Mumbai, India
University of Mumbai

pratibha.deshmukh@bharativedyapeeth.edu

²Dept. of Centre for Continuing Education,
Indian Institute of Science (IISc), India
chhetri.com@gmail.com

³Dept. of Computer Science & Engineering,
School of Engineering and Technology,
Dean – CSE, Apex University
Jaipur, India

sh_avinash@yahoo.com; avinash.sharma@apexmail.in;

Orchid ID: 0000-0002-4602-9728

⁴Dept. of Computer Science & Eng., School of Eng. & Tech.
CGC University, Mohali, Panjab, India
rini.j1099@cgc.ac.in

⁵Bharati Vidyapeeth's Institute of Management and Information
Technology Navi Mumbai, India
University of Mumbai
yogeshgangarde12@gmail.com

⁶Department of Computer Applications
Sharda University Agra, India
nidhi4407@gmail.com

Abstract

Traditional land registration systems are highly vulnerable to document forgery, fraud, and inefficiencies due to centralized control, leading to various digital crimes and system inefficiencies. In this paper, we have proposed the design and implementation of a blockchain-based land registry framework utilizing a private permissioned blockchain architecture. Our system integrates smart contracts for property registration, ownership transfer, and user verification, along with the InterPlanetary File System (IPFS) for decentralized storage of legal documents. The proposed framework follows a structured methodology comprising system design, implementation, deployment, testing, and validation. We have integrated the necessary technology stack, which includes Solidity, Ethereum, MetaMask, and IPFS, to ensure secure and transparent transactions. Our system exhibits improved efficiency, immutability, and trust compared to traditional land registry mechanisms in use. Further, we have performed

the Functional validation using black-box testing, which confirms the robustness of the framework. The obtained results indicate that permissioned blockchain systems have significant potential to transform land registration processes by ensuring transparency, reducing intermediaries, and enhancing operational reliability.

Keywords - Blockchain, Land Registry, Smart Contracts, IPFS, Decentralization, Ethereum, Property Management

1. Introduction

In the current landscape of real estate, especially in developing and underdeveloped countries, the lack of technology infrastructure and practices leads to a lot of issues in land registries. Traditional land registries often resemble fragile relics centralized vaults where precious property papers are vulnerable to systemic failure, corruption, or physical loss. When citizens purchased land, government authorities would grant them by uploading digital copies to IPFS. Thus, a decentralised fortress is immune to single points of collapse. Smart contracts acting as digital guardians, validated each document's authenticity through ethereum blockchain. Moreover the chain of ownership where property rights could never be tampered [1]. Buying or selling property leads to navigating of fraud, delays and a system bleeding trust and efficiency [2]. The land registration process is complicated and time-consuming and required a lot of effort to transfer ownership. Blockchain is like a database wherein anyone with authorization to access documents can secure such events using cryptographic algorithm [3], [4]. Use of smart contracts determines the owner, seller, and the transfer of the land. A distributed ledger system that maintains a track of each of the transactions. It uses a peer-to-peer network architecture [5], [6].

Additionally need for efficiency in land registration faced by policymakers and practitioners at large. Creating records for the physical assets into an immutable form is possible using Blockchain [7], [8]. In this regard transparent land registration system that protects property rights and prevents data manipulation [9], [10]. Contemporary land registry systems plagued by corruption, human error, and obsolete infrastructure. Also, face critical challenges in tracking property ownership. Blockchain attracts governments globally by enabling direct legal transfer of land ownership without third-party verification [11], [12], [13]. The proposed system employs cryptographic hashing (SHA-256) and decentralized storage in IPFS for owner registration and document submission, ensuring data integrity and tamper resistance. Smart contracts developed in Solidity make land registration and ownership transfer processes transparent and automated. At the same time, using modern web technologies like React and Node.js increases the accessibility of the system and interaction with the user [14][15][16]. Several studies in different countries indicate the fact that traditional digital land registry systems are still vulnerable to forgery of documents, data tampering, and intermediary control . Further, limited transparency combined with time-consuming verification procedures are sources of operational inefficiency.

[17][18][19]. These challenges are consistent across regions, though the implementation approaches may differ. Smart contracts solve these problems, enabling consistent verification, the removal of intermediaries and increasing the transparency of transactions [20][21]. Previous studies have shown that the traditional land registration process often takes weeks to months to complete, and asset transfers involve a lot of procedural complexity. These systems are also affected by the lack of transparency, security problems and lack of accountability. To deal with these challenges, implementation of blockchain technology is a promising solution that ensures data transparency, immutability and tamper resistance. In this way, blockchain-based systems can efficiently eliminate inefficiencies and boost trust in land transactions. In this sense, the present work proposes and demonstrates a robust blockchain-based architecture for a secure and efficient management of land registries.

2. Materials and Methods

The study follows a systematic approach that consists of literature review, system design, implementation, testing and a comparative study. A comprehensive literature review was conducted to examine existing land registry systems and blockchain-based frameworks. This involved analysing peer-reviewed journals and conference proceedings. Based on the identified gaps a blockchain-based land registry architecture designed.

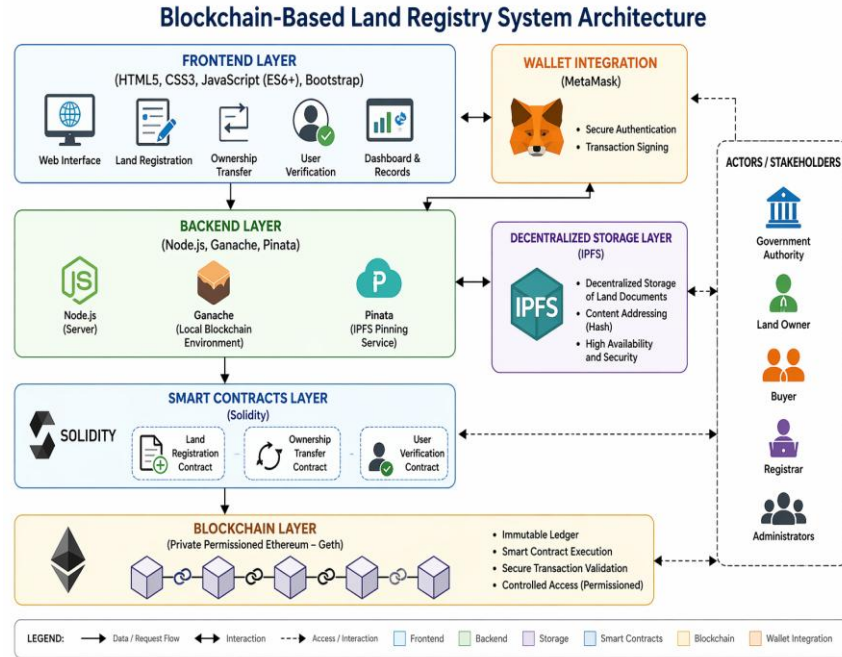


Fig 1. Proposed system design for the blockchain-based land registry system.

The proposed blockchain-based land registry system employs a layered architecture to provide security, transparency, and efficiency in property management. It employs blockchain technology, decentralized storage, and smart contract automation to address the limitations of traditional systems.

- **Blockchain Layer:** A private permissioned blockchain ensures secure transaction validation and controlled access, while maintaining an immutable ledger of land records.
- **Decentralized Storage Layer:** Land documents stored in a distributed manner using IPFS and cryptographic hashes, ensuring data integrity and availability.
- **Smart Contract Layer:** Smart contracts written in Solidity automate the process of land registration, transfer of ownership and verification of users, removing the need for intermediaries and building trust.
- **Frontend Layer:** Developed using HTML5, CSS3, JavaScript and Bootstrap for a user friendly interface for interacting with the system.
- **Backend Layer:** Created in Node.js with Ganache for testing and IPFS integration via Pinata for seamless communication across the system.
- **Blockchain Platform:** Based on private Ethereum (Geth) and Hyperledger Fabric as alternative for the enterprise scalability.
- **Wallet Integration:** MetaMask provides secure authentication and transaction signing.
- **Security Mechanism:** Only authenticated users can access, enabling secure and verified transactions.
- **Testing & Validation:** End-to-end validation and black-box testing ensure correct functionality and transfer of ownership in the proper manner.

III. Transferring of Assets

This diagram describes, secure way to transfer unique digital ownership in terms of assets.

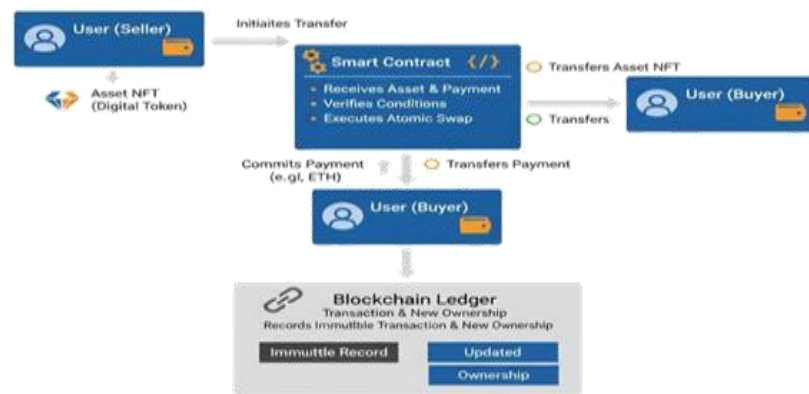


Fig. 2. Transferring of Assets

Algorithm 1.1: Transfer of agents

Input: Seller S, Buyer B, Asset ID A

Output: Updated ownership record on blockchain

Start

Step 1: Verify Seller Identity (S)

Step 2: Check ownership of Asset A by Seller S

Step 3: Seller initiates transfer request via smart contract

Step 4: Buyer B sends payment to smart contract

Step 5: Smart contract validates:

a) Seller is the legitimate owner

b) Buyer has sufficient funds

Step 6: If validation successful:

a) Transfer ownership of Asset A to Buyer B

b) Transfer payment to Seller S

Step 7: Record transaction on blockchain ledger

Step 8: Generate transaction confirmation

End

The asset transfer mechanism is performed by a smart contract. This allows for a secure and automated transfer of ownership. The seller first calls the smart contract to ask for a transfer. The smart contract verifies that the seller is the owner of the asset. If verified successfully, the buyer pays the smart contract. Then the smart contract does a conditional check to make sure the seller is the actual owner and the buyer has enough money. Once the conditions are fulfilled the contract atomically performs the transfer updating ownership records and releasing the payment to the seller. Finally, the transaction is permanently recorded on the blockchain ledger, ensuring immutability, transparency and traceability. The automation removes intermediaries, reduces fraud and builds trust in the land registry system. For better understanding we have given the technical details in Algorithm 1.1 and Figure 2. For the asset transfer mechanism.

3. Mathematical Modelling of the Framework

To formalize the security and operational integrity of the proposed land registry system, we define the following mathematical models for data integrity and decentralized storage.

Data Integrity and Hashing

Every land record L consists of property details and legal documentation. To ensure immutability, we apply a cryptographic hash function H (specifically SHA-256). The state of a record at time t is represented as:

$$S_t = (P_i, O_i, D_{ipfs})$$

Where:

- P_i represents unique property identifiers.
- O_i represents the owner's cryptographic public key.
- D_{ipfs} is the Content Identifier (CID) of the documents stored on IPFS.

Smart Contract State Transition

The transfer of ownership is an atomic state transition function σ executed by the smart contract. Given a seller S , a buyer B , and an asset A , the transition is defined as:

$$\sigma(S, B, A) = \{Q_{new} = B, Q_{new} = S, \text{if } Valid(S, A) \wedge Funds(B) \geq Price \text{ otherwise}\}$$

The validity function $Valid(S, A)$ checks the blockchain ledger to confirm that the seller's address S matches the current owner record for asset A .

IPFS Content Addressing

The decentralized storage layer utilizes a Distributed Hash Table (DHT) to locate files. The probability of document loss $P(loss)$ in the IPFS network with n nodes and a replication factor r is modeled as:

$$P(loss) = (1 - P)^{n \cdot r}$$

Where P is the probability of a single node being online and accessible. By utilizing Pinata for IPFS pinning, we maximize P , ensuring near-constant availability of legal deeds.

4. Implementation of Land Registry Framework

The Smart Contracts for land registration automated with all processes including Metamask Wallet connection along with IPFS.

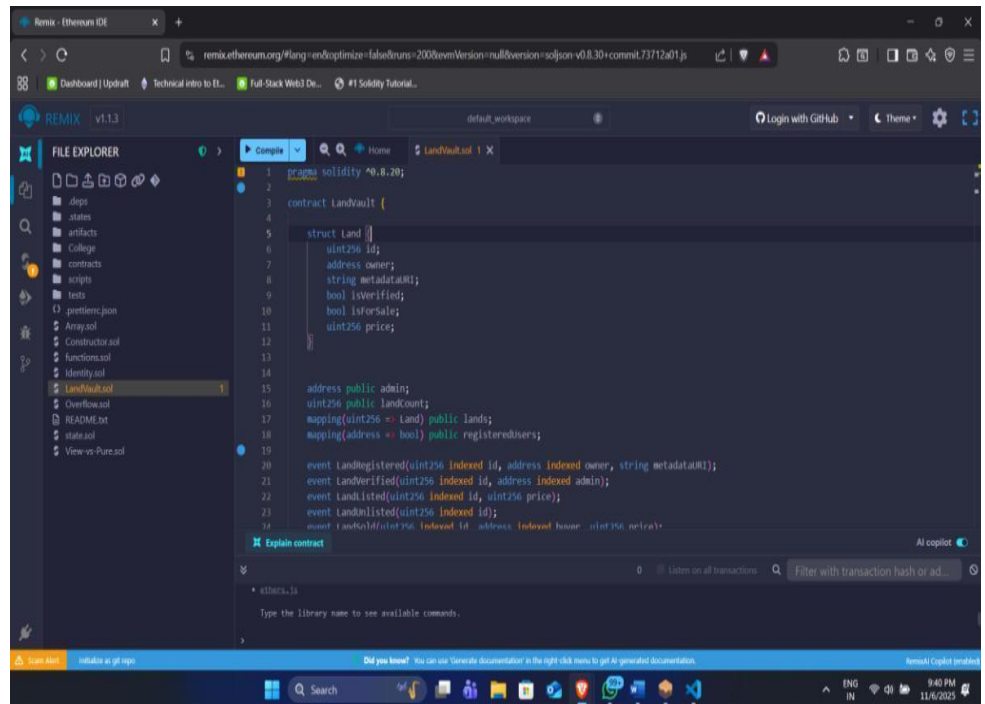


Fig. 4.1 Smart Contract for land registry

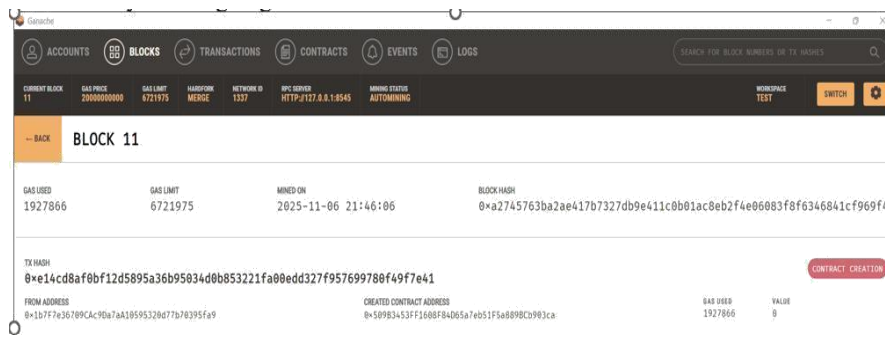


Fig. 4.5 Deployed smart contract in ganache.

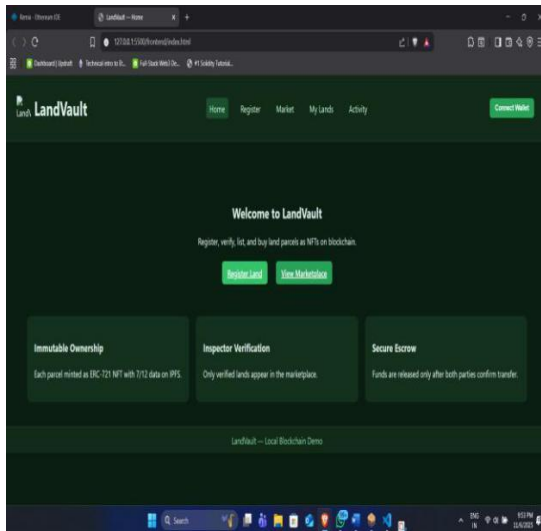


Fig.4.6 Landing page of landvault

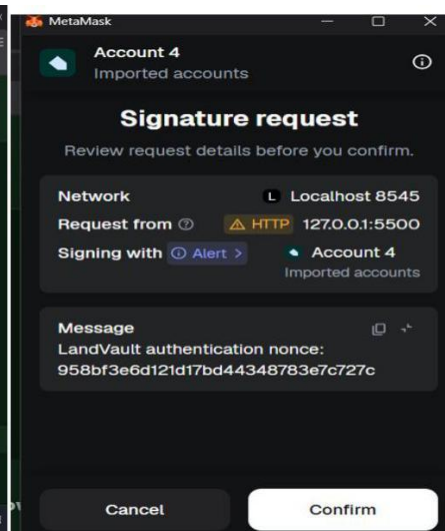


Fig.4.7. Metamask wallet connection.

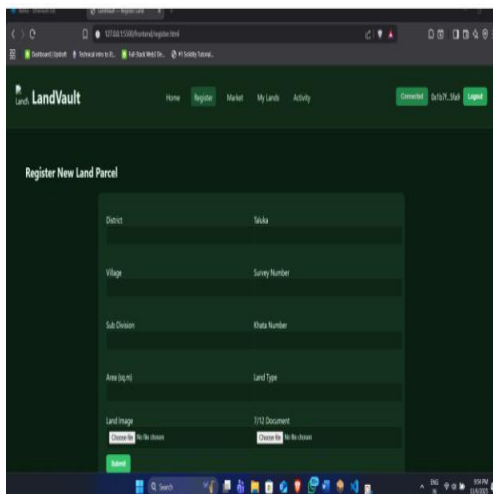


Fig.4.8 Land registration-Landvault

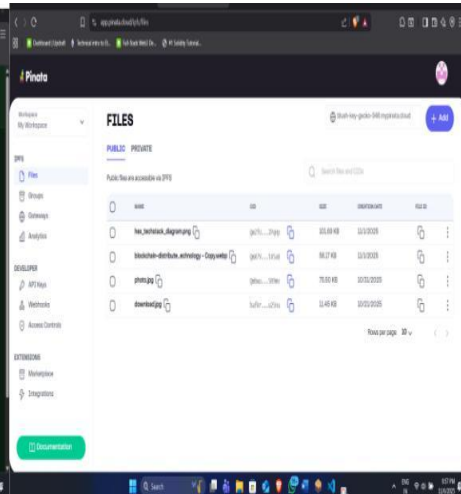


Fig.4.9. Uploaded files in Pinata IPFS

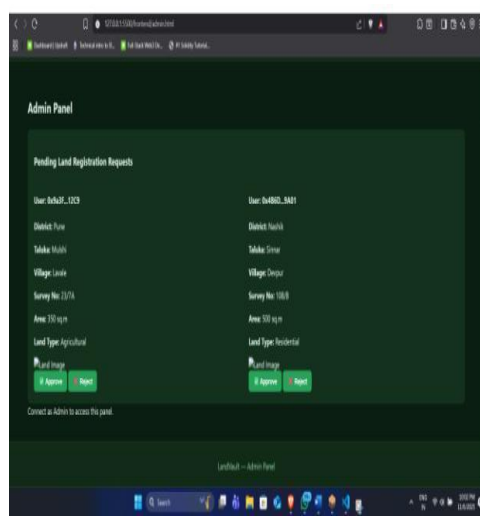


Fig.4.9 Admin dashboard of landvault



Fig. 4.10 Land selling page of landvault

5. Experimental Results and Discussion

The proposed framework was implemented using Solidity for smart contracts and Ganache as a local blockchain environment. Functional validation was performed using black-box testing to ensure the robustness of the asset transfer logic.

Performance Evaluation

The system's efficiency was measured against that of traditional centralised databases. While traditional systems involve manual verification that can take weeks, our smart contract-based automation executes transfers in seconds once the transaction is mined.

Parameter	Traditional System	Proposed Framework
Data Storage	Centralized (Vulnerable)	Decentralized (IPFS + Ledger)
Verification	Manual / Intermediary	Automated (Smart Contract)
Immutability	Low (Susceptible to forgery)	High (Cryptographic Hashing)
Transparency	Limited	Publicly Verifiable (Permissioned)
Process Time	Weeks / Months	Near Real-Time

Security Analysis

The integration of MetaMask for wallet-based authentication ensures that only authorized entities can initiate transactions. During testing, we verified the following security properties:

- **Tamper Resistance:** Any change to the legal document D on IPFS results in a completely different CID.
- **Double-Spending Prevention:** The smart contract ensures a property cannot be sold to two buyers simultaneously.

Discussion

In this paper we have presented the framework for the transition from a centralized, paper-dependent registry to a decentralized, blockchain-based framework represents more than just a technological upgrade; this will serve as fundamental shift toward digital sovereignty and trust. We leveraged Private Permissioned Blockchain architecture, which provides a robust solution to the "trust deficit" often found in real estate transactions, particularly in developing regions where document forgery and data tampering are prevalent. This work also presented integration of IPFS, ensuring that legal deeds are not only immutable but also highly available, effectively eliminating the risk of physical loss due to systemic failure or corruption. Furthermore, we have also provided automation by Solidity-based smart contracts, which reduces the reliance on costly and time-consuming intermediaries, thereby lowering the barrier for property ownership and ensuring that transactions are executed with mathematical certainty. The proposed framework demonstrates that by combining cryptographic security with modern web accessibility, we can create a land administration system that is inherently transparent, auditable, and resilient to the challenges of the digital age.

6. Conclusion

The research proposes a blockchain-based land registry framework to improve transparency, security, and efficiency in property management. The proposed system combines a private permissioned blockchain with smart contracts and decentralized storage (IPFS) to provide immutable record-keeping without the need for intermediaries. Smart contracts are used to automate land registration and ownership transfer, which saves a lot of time in processing and reduces the risk of fraud. The implementation and testing of the system show that it is feasible and effective for the processing of secure transactions and ensuring the integrity of data. Wallet-based authentication strengthens access control and provides trusted interactions among stakeholders. The proposed framework provides reliability, auditability and operational efficiency over the traditional land registry systems.

In conclusion, the study highlights the game-changing nature of blockchain technology in revolutionizing land administration systems. Future work may include the integration of advanced features such as AI-based fraud detection, GIS mapping, and large-scale deployment for increased scalability and real-world applicability.

References

1. D. Shinde, S. Padekar, S. Raut, A. Wasay and S. S. Sambhare, "Land Registry Using Blockchain - A Survey of existing systems and proposing a feasible solution," 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), Pune, India, 2019, pp. 1-6, doi: 10.1109/ICCUBEA47591.2019.9129289.
2. M. Biswas, J. A. Faysal and K. A. Ahmed, "LandChain: A Blockchain Based Secured Land Registration System," 2021 International Conference on Science & Contemporary Technologies (ICSCST), Dhaka, Bangladesh, 2021, pp. 1-6, doi: 10.1109/ICSCST53883.2021.9642505.
3. R. K. Yadav, R. Dabare, M. Ghyar, S. Bhongade and M. Gautam, "Smart Contract-Based Land Registration System Using Blockchain," 2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 2023, pp. 1-6, doi: 10.1109/SCEECS57921.2023.10063068.

4. A. Riaz, M. J. Ikram and N. Asadullah, "Using Blockchain to Overcome the Issues in Land Registry Management: A Systematic Review," 2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC), Jeddah, Saudi Arabia, 2023, pp. 1-6, doi:10.1109/ICAISC56366.2023.10085673.
5. S. K. Singh, A. Tiwari, P. R. Singh, N. Singh and P. Kumar, "Land Registration System using Blockchain," 2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2023, pp. 1499-1503, doi: 10.1109/ICAC3N60023.2023.10541693.
6. M. Kotkar, V. Ahuja, D. Kharode and K. Lahoti, "Land Registration System Using Blockchain – A Decentralized Application with IPFS & Ethereum," 2023 IEEE Fifth International Conference on Advances in Electronics, Computers and Communications (ICAECC), Bengaluru, India, 2023, pp. 1-5, doi: 10.1109/ICAECC59324.2023.10560138.
7. A. K. Mahto, "Blockchain Technology for Secure and Efficient Land Registration Management: A Promising Paradigm Shift in India," 2023 International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS), Erode, India, 2023.
8. M. Nandi, R. K. Bhattacharjee, A. Jha and F. A. Barbhuiya, "A secured land registration framework on Blockchain," 2020 Third ISEA Conference on Security and Privacy (ISEA-ISAP), Guwahati, India, 2020, pp. 130-138, doi: 10.1109/ISEA-ISAP49340.2020.235011.
9. R. C. Suganthe, N. Shanthi, R. S. Latha, K. Gowtham, S. Deepakkumar and R. Elango, "Blockchain enabled Digitization of Land Registration," 2021 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2021, pp. 1-5, doi: 10.1109/ICCCI50826.2021.9402469.
10. S. Humdullah, S. Hajar Othman, M. Najib Razali and H. Kutty Mammi, "Secured Data Storage Framework for Land Registration using Blockchain Technology," 2021 3rd International Cyber Resilience Conference (CRC), Langkawi Island, Malaysia, 2021, pp. 1-6, doi: 10.1109/CRC50527.2021.9392434.
11. I. Mishra, Supriya, A. Sahoo and M. Vivek Anand, "Digitalization of Land Records using Blockchain Technology," 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2021, pp. 769-772, doi: 10.1109/ICACITE51222.2021.9404678.
12. S. Verma, S. Kumar Pathak, P. Chaudhary, P. Yadav and S. Yousuf, "Land Registry System Using Blockchain Technology," 2024 International Conference on Emerging Innovations and Advanced Computing (INNOCOMP), Sonipat, India, 2024, pp. 170-177, doi:10.1109/INNOCOMP63224.2024.00035.
13. S. Suresh, K. Van, C. Yadav, M. Jacob and V. Wangikar, "Cadastral: Blockchain-Based Land Registration System," 2024 11th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Yogyakarta, Indonesia, 2024, pp. 320-325, doi:10.1109/EECSI63442.2024.10776335.
14. S. Dubey, D. Vinod, A. Gupta and R. K. Dwivedi, "Secure Land Registration Management via Ethereum Blockchain," 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 2023, pp. 185-191, doi: 10.1109/IDCIoT56793.2023.10053494.
15. A. L. Shrivastava and R. Kumar Dwivedi, "Blockchain-based Secure Land Registry System using Efficient Smart Contract," 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 2023, pp. 165-170, doi: 10.1109/IDCIoT56793.2023.10053476.
16. R. Sidharthan and V. R. Balasaraswathi, "Secured Land Registration using Ethereum Blockchain and IPFS," 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2024, pp. 1449-1454, doi: 10.1109/ICAAIC60222.2024.10575276.
17. K. Samal, B. K. Mohanta, S. Sharma and D. Jena, "Secure Digitization of Land Record using Blockchain Technology in India," 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2021, pp. 1-7, doi: 10.1109/ICCCNT51525.2021.9579946.
18. Y. Borse, P. Ahirao, A. Chawathe and D. Patole, "Embracing Permissioned Blockchain for a Secure and Reliable Land Registration System," 2022 5th International Conference on Advances in Science and Technology (ICAST), Mumbai, India, 2022, pp. 525-530, doi:10.1109/ICAST55766.2022.10039485 }.
19. M. Stefanović, S. Ristić, D. Stefanović, M. Bojkić and D. Pržulj, "Possible Applications of Smart Contracts in Land Administration," 2018 26th Telecommunications Forum (TELFOR), Belgrade, Serbia, 2018, pp. 420-425, doi: 10.1109/TELFOR.2018.8611872.
20. A. Sahai and R. Pandey, "Smart Contract Definition for Land Registry in Blockchain," 2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT), Gwalior, India, 2020, pp. 230-235, doi: 10.1109/CSNT48778.2020.9115752.
21. A. F. Mendi, K. K. Sakaklı and A. Çabuk, "A Blockchain Based Land Registration System Proposal for Turkey," 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Istanbul, Turkey, 2020, pp. 1-6, doi: 10.1109/ISMSIT50672.2020.925507.
22. Deshmukh, P., Avinash, S., Gonsai, A.M., Sonawane, S.S., Khan, T. (2023). Ordering Services Modelling in Blockchain Platform for Food Supply Chain Management. In: Rathore, V.S., Tavares,

- J.M.R.S., Piuri, V., Surendiran, B. (eds) Emerging Trends in Expert Applications and Security. ICE-TEAS 2023. Lecture Notes in Networks and Systems, vol 681. Springer, Singapore. https://doi.org/10.1007/978-981-99-1909-3_37
23. Prof. Pratibha M. Deshmukh, Hemant Patil, Dr. Pallavi Jamsandekar (2019), BLOCKCHAIN MODEL FOR CANE DEVELOPMENT WITH RESPECT TO FARMERS, SSN: 1847-9790, Volume: 03 Issue: 04
24. P. Deshmukh, J. Kharade, A. Sharma, A. Pawar, S. Gharat and R. Saxena, "Designing a Framework for Electric Vehicle Ecosystems in India with Respect to Cybersecurity," 2025 2nd International Conference on Computational Intelligence and Computing Applications (ICCICA), Samalkha, India, 2025, pp. 753-759, doi: 10.1109/ICCICA67008.2025.11337322.