WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

SECURE GENOME ANALYTICS AND TRADE NETWORK

Rajesh M[1], Aditya Ashirwad[2], Aditi Rathi[3], Chandan K L[4]

[1] Assistant Professor Dept. of CSE, DSATM
[2] Dept. of CSE, DSATM adityaashirwad99@gmail.com
[3] Dept. of CSE, DSATM aditirathi5656@gmail.com
[4] Dept. of CSE, DSATM chandankl1015@gmail.com

ABSTRACT—

With the exponential growth of genomic data, efficient and intelligent systems are crucial for managing, predicting, and monetizing genomic datasets. This project presents a novel platform that integrates machine learning, web technologies, and blockchain to predict genomic scaffold counts based on metadata and facilitate their secure trade. The system utilizes a Random Forest and Neural Network model to estimate scaffold counts using features such as accession length, organism type, assembly level, and release date. These predictions inform the pricing of genomic data, which is then listed and bid on using Ethereum-based smart contracts. A user-friendly Streamlit dashboard interacts with a Flask API and a deployed smart contract to provide a seamless and trustworthy experience for researchers, labs, and data contributors. This unified approach empowers both scientific advancement and decentralized commerce in the genomics domain.

KEYWORDS— Secure Genome Analytics And Trade Network , Blockchain, Smart Contracts, Token based Incentives, Data integrity, Data security

INTRODUCTION

The rise of next-generation sequencing has generated a deluge of genomic data, creating an urgent need for tools that not only analyze but also manage and monetize such information. Genomic scaffolds, which represent assembled fragments of DNA sequences, are vital in understanding genetic structures, variations, and evolutionary patterns. Accurately predicting the number of scaffolds in an assembly helps researchers estimate the completeness and complexity of the genome.
In parallel, blockchain technology has emerged as a robust mechanism for enabling secure, transparent, and decentralized transactions, including the exchange of digital data assets. Combining the predictive power of machine learning with the transparency of blockchain provides a groundbreaking solution for the genomic data marketplace.

This project bridges these technologies by building a genomic scaffold predictor and marketplace. Machine learning models provide scaffold count predictions from genomic metadata, while a smart contract governs the creation, bidding, and sale of data listings. A Flask backend handles prediction logic, and a Streamlit frontend serves as a user-friendly dashboard for interaction and visualization.

This platform leverages predictive modeling to estimate genomic scaffold counts using curated metadata and empowers users to trade these datasets transparently using blockchain. By combining AI and decentralized technologies, the system promotes data accessibility, trust, and monetization in the growing field of genomics.

RELATED WORK

Machine Learning for Genomic Prediction:

Machine learning has significantly influenced bioinformatics, particularly in the prediction of genome-related metrics. Random Forest and Neural Networks have been utilized in genomics to model relationships between genomic features and biological outcomes. [1] Breiman (2001) introduced Random Forest as an ensemble method that is resilient to overfitting and robust to noisy data, making it suitable for complex biological datasets. Similarly, deep learning approaches such as those reviewed by [2] Min et al. (2017) have demonstrated success in learning representations directly from DNA sequences for tasks such as promoter and enhancer prediction.

For scaffold count prediction, the work of Parks et al. (2015)[3] on CheckM introduced a method to estimate genome completeness and contamination using marker genes, which indirectly relates to scaffold count estimation. Although not directly predicting scaffold count, these models indicate the feasibility of metadata-driven genomic quality assessment.

Blockchain Applications in Genomic Data Sharing :

Blockchain has been proposed as a solution to the privacy, security, and transparency challenges in genomic data sharing. Research by Nugent et al. (2016)[4] proposed a blockchain-based framework for securely sharing genomic data, ensuring traceability and tamper-resistance. In similar work, Pattengale et al. (2020)[5] demonstrated the use of smart contracts for permissioned access to genomic datasets on Ethereum.

These studies highlight the potential of decentralized systems
in enabling trustless data exchange—a feature that this project adopts in listing and bidding for genomic records via smart contracts.

Genomic Data Marketplaces and Incentivization :

Emerging platforms like Genomes.io, EncrypGen, and Nebula Genomics emphasize user-controlled, monetizable genomic data. Erlich et al. (2018)[6] argued for secure genomic data markets to allow individuals to profit from their data while maintaining consent and control. Furthermore, Kuo et al. (2017)[7] conducted a comprehensive review of blockchain applications in healthcare, highlighting how smart contracts can automate transactions and enforce policies.

Our project draws from these innovations, incorporating predictive pricing and secure exchange mechanisms directly into the system.

SYSTEM ARCHITECTURE

1. *Frontend Layer — Streamlit Dashboard*

The frontend is implemented using Streamlit, a lightweight Pythonbased web application framework optimized for data science and machine learning applications.

User Interface (UI):
Users interact through a clean, responsive UI where they can input genomic metadata (Organism type, Assembly Level, Accession Length, Release Date) in a guided and validated format (select boxes, number inputs, and date pickers).

Prediction Interface:
A "Predict via Flask API" button triggers scaffold count predictions. The dashboard communicates with the backend API through HTTP POST requests and displays predictions from both machine learning models instantly to the user.

Listing and Bidding Interface:
Users can create new listings based on prediction results by providing a unique Accession ID and setting an initial price (in Ether). They can also place bids on existing listings or finalize a sale after bidding, ensuring full blockchain interaction through a simple interface.

Model Evaluation Visualization:
Users can expand a section to see the model's performance using scatter plots, real vs predicted graphs, and error metrics (like Mean Absolute Error), helping build trust in the predictions.

Web3 Connectivity:
The frontend seamlessly interacts with Ethereum smart contracts using web3.py, signing transactions, submitting bids, and finalizing sales with minimal user effort.

2. *Backend Layer — Flask API Server*

The backend is powered by a lightweight Flask server that acts as an intermediary between the user interface and the machine learning models.

Request Handling:
When a prediction request is initiated, the frontend sends a JSON payload to the Flask server containing the user's input features.

Model Loading:
Pretrained models (Random Forest and Neural Network) are loaded from serialized .pkl files (joblib) into memory for fast inference.

Prediction Processing:
The server processes the incoming feature vector, reshapes it as required, and sends it through both models to generate predictions.

Suggested Pricing Logic:
A pricing logic module suggests an initial sale price based on the Random Forest prediction, typically using a heuristic like scaling by a factor (e.g., 0.0001 ETH per scaffold predicted).

Response Formatting:
The Flask server sends back a wellstructured JSON response including:Random Forest Prediction, Neural Network Prediction, Suggested Price for Blockchain Listing

Error Handling:
Robust trycatch mechanisms ensure that even if a model inference fails, appropriate HTTP error responses are sent.

3. *Machine Learning Layer — Model Inference and Prediction*

Two robust models are employed to ensure predictive performance across diverse genomic datasets:

Random Forest Regression Model:

Trained to handle both numerical and categorical features. Provides a stable baseline with good interpretability.Resilient to overfitting, especially with mediumsized datasets.

Neural Network Regression Model:

Built using TensorFlow/Keras.Captures complex, nonlinear interactions between input features. Trained with Mean Squared Error (MSE) loss and early stopping for optimal performance.

Both models were trained offline using a carefully preprocessed genomic dataset including accession metadata and normalized numerical features, then serialized and loaded into production.

4. *Blockchain Layer — Smart Contract System*

At the heart of the marketplace is a Solidity based smart contract deployed on a local Ethereumcompatible blockchain (Ganache/Hardhat).

Smart Contract Features:

createListing(string accession, uint scaffoldCount, uint startingPrice): Allows users to create a listing with scaffold predictions and a starting price. **placeBid(string accession):**Enables users to place a bid on an existing listing.Bids must be higher than current price. **finalizeSale(string accession):** Finalizes the sale to the highest bidder and completes ownership transfer.

Security Features:

Transaction signing and submission handled via Web3.Gas fee management and nonce management to avoid transaction errors. Protection against low bids or unauthorized finalization.

Web3.py Integration:

Transactions are built, signed with a private key, and submitted programmatically using web3.py, ensuring full decentralization without exposing blockchain complexity to users.

5. *Overall Workflow*

a) User Input Phase:

User opens Streamlit dashboard.User fills in organism metadata and hits "Predict".Input features are packaged into a JSON request.

b) Prediction Phase:

Flask server receives input and triggers ML models.Scaffold count predictions are generated and sent back.

c) Listing Phase:

Users can create listings with accession IDs and suggested or custom prices.Listings are registered immutably on the Ethereum blockchain.

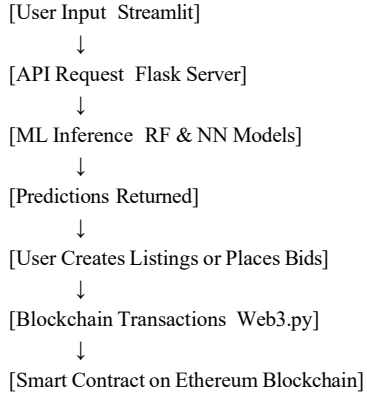
d) Bidding Phase:

Other users can view available listings.Bidders submit higher offers to win the dataset.All bids are securely stored onchain.

e) Finalization Phase:

Original listers finalize sales.Ownership and financial transactions are securely closed via blockchain.

6. *Data and Control Flow Diagram*



METHODLOGY

The development of the **Genomic Scaffold Predictor & Marketplace** system followed a modular and iterative engineering methodology encompassing data preprocessing, model training, system integration, and smart contract deployment. The core objective was to create a predictive and transactional platform for genomic datasets using machine learning and blockchain technologies. The entire methodology is divided into the following key phases:

1. **Data Acquisition & Preprocessing**

The project utilized a real-world genomic dataset containing metadata such as: Accession ID,Organism Name,Assembly Level,Release Date,Scaffold Count

Key Preprocessing Steps:

Missing Value Handling: Rows with missing or null values were removed.Label Encoding: Categorical fields like *Organism* and *Assembly Level* were encoded using LabelEncoder.Date Transformation: The *Release Date* field was converted into a numeric "days since release" value.Feature Engineering: Additional features like accession length and encoded assembly levels were introduced.Normalization: Numerical features were scaled using MinMaxScaler to bring values within a standard range for ML models.

2. **Model Design and Training**

Two machine learning models were trained using the preprocessed dataset:

A. Random Forest Regression: Implemented using scikit- learn.Trained on structured features to predict scaffold count.Robust to outliers and efficient on small-to-medium datasets.

B. Neural Network Regression: Implemented using Keras (TensorFlow backend).Architecture: 3 dense layers with ReLU activations and an output layer with linear activation.Trained using MSE loss function and Adam optimizer with early stopping to prevent overfitting.Both models were trained and evaluated on the same dataset, and serialized using joblib for real-time inference.

3. API Development (Backend Layer): A lightweight Flask API was built to serve the trained models.Input: JSON requests containing the encoded and normalized metadata features.Output: JSON responses with predicted scaffold counts from both models, and a suggested price.Error Handling: Included structured responses for invalid requests or inference failures.Deployment: Hosted locally or via container for seamless integration with the frontend.

4. Frontend Interface (Streamlit Dashboard): Streamlit was used to create an interactive, elegant UI with the following components:

Input Panel: Allows user to enter organism metadata and accession details.

Prediction Panel: Displays results from both models and a .finalizeSale(): Transfers ownership to the highest bidder after closing.

Web3 Integration: web3.py was used to connect Streamlit to the blockchain.Private keys and account addresses were securely managed.Transactions were signed, submitted, and tracked via Web3's transaction receipt system.

5. **Workflow Integration**

Each component was integrated into a seamless workflow:

User inputs metadata → Streamlit, Data sent → Flask API for prediction,Predicted scaffold count → shown to user, User creates listing → Web3 sends to contract,Other users bid → on-chain transactions, Sale finalized → ownership transferred on-chain.

6. **Testing and Validation**

Model Validation: Performed using cross-validation and Mean Absolute Error (MAE).

Unit Testing: Implemented for API endpoints and contract functions.**Smart Contract Testing**: Performed using Hardhat + JavaScript test cases.**Manual Testing**: UI functionality and error scenarios were tested through repeated runs.

7. Tools & Technologies Used Layer Tool/Tech

Python, Pandas, Scikit-learn,
Implemented in Python using Pandas, NumPy, and scikitlearn, the pipeline transforms raw genomic metadata into structured inputs. Key Implementation Steps:
CSV Loading: pandas.read_csv() for importing raw data.
Label Encoding: LabelEncoder() to convert categorical values like organism and assembly level into integers.
Date Parsing: Using datetime.strptime() and subtraction to convert release dates into a numeric delta (days since release).
Feature Engineering: Fields like accession length and level length were manually computed.
Scaling: MinMaxScaler() was applied to normalize feature values between 0 and 1.
This processed dataset was saved for training and realtime predictions.

2. Machine Learning Models

Two regression models were implemented:

A. Random Forest Regressor

Library: sklearn.ensemble.RandomForestRegressor
Parameters: Tuned with grid search on n_estimators, max_depth, and min_samples_split.
Exported via: joblib.dump() to genomic_rf_model.pkl.

B. Neural Network Regressor

Library: TensorFlow / Keras Layers:
Input layer: 5 neurons
Hidden layers: [64, 32] with ReLU activation Output layer: 1 neuron (regression) Optimizer: Adam
Loss: Mean Squared Error
Exported via: model.save("genomic_ai_model.h5")
Both models were trained on the same dataset and evaluated using
Data & ML
TensorFlow/Keras
mean_absolute_error().
Backend Flask, Joblib, JSON Frontend Streamlit
Blockchain Solidity, Hardhat, Ganache, Web3.py Visualization Matplotlib, Streamlit Charts
Deployment Localhost testing with potential for cloud/container deployment

IMPLEMENTATION DETAILS

The Genomic Scaffold Predictor & Marketplace project integrates machine learning with decentralized blockchain technology in a unified application. The implementation was divided into modular components to ensure maintainability, scalability, and ease of integration. This section elaborates on the technical details of each layer and component of the system.

1. Data Preprocessing Pipeline

Flask API Backend

A lightweight Flask server acts as a prediction microservice. Features: Endpoint: /predict, Method: POST, Input: JSON with 5 numeric features. Output: JSON response with predictions from both models and a derived ETH price.

Python:

```
@app.route('/predict', methods=['POST']) def predict():
    data = request.json['features']  rf_pred = rf_model.predict([data])[0]
    nn_pred = nn_model.predict(np.array([data]))[0][0] price = float(np.log1p((rf_pred + nn_pred) / 2))  0.001 return jsonify({
        "RandomForest": rf_pred, "NeuralNet": nn_pred, "SuggestedPrice": round(price, 5)
 })
```

3. Streamlit Frontend Interface

Built using Streamlit, the UI is interactive and userfriendly. Functionalities:

Metadata Input: Uses st.selectbox(), st.number_input(), st.date_input().Model Inference: Sends user inputs to the Flask API using requests.post().Visualization: Displays prediction metrics and scatter plots with matplotlib.
Blockchain Interaction: Integrated using web3.py for listing, bidding, and finalizing.Streamlit's CSS was customized for a sleek, modern feel using st.markdown() with inline HTML styles.

4. Smart Contract Deployment

The Solidity smart contract was compiled and deployed using Hardhat.
Contract Highlights:function createListing(string memory accession, uint scaffoldCount, uint price) public {...} function placeBid(string memory accession) public payable. function finalizeSale(string memory accession) public {...}

5. Web3 Integration (Python)

Used web3.py to connect the frontend (Streamlit) to the Ethereum blockchain.
Steps:
Connected via HTTPProvider("http://127.0.0.1:8545") Loaded contract ABI using json.load()
Read and write contract state using:
.functions.functionName().call()
.functions.functionName().build_transaction()
Example: txn = contract.functions.createListing(accession, predicted, w3.to_wei(price, 'ether')).build_transaction({...}) signed = w3.eth.account.sign_transaction(txn, private_key=PRIVATE_KEY)w3.eth.send_raw_transaction(signed.rawTransaction)

6. Security & Validation

User Input Validation: Handled on the frontend to ensure complete inputs.
Private Key Handling: Local key storage; to be replaced by wallet integrations like MetaMask for production.
Contract Constraints: Enforced bid amounts to exceed current price; restricted finalization to contract owner.

7. Directory Structure GenomicMarketplace

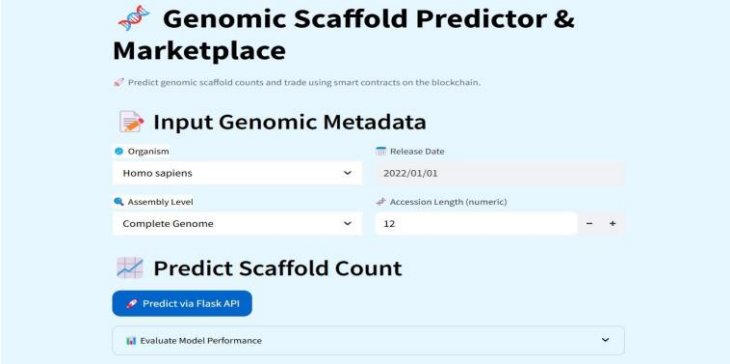
```
├── dashboard.py                Streamlit frontend
├── api.py                      Flask backend
├── artifacts/
│   └── GenomicMarketplace.sol/
├── models/
│   ├── genomic_rf_model.pkl
│   └── genomic_ai_model.h5
├── processed data/
│   └── final_processed_data.csv
├── test/
│   └── contract_test.js
└── utils/
    └── encoder_scaler.pkl
```

RESULTS

The performance of the Genomic Scaffold Predictor & Marketplace was evaluated through two key dimensions: predictive accuracy of the machine learning models and functional correctness of the blockchain integrated marketplace. This section summarizes both qualitative and quantitative outcomes of the system.

1. **Input Data:**

The genomic data of Homo sapiens collected on the specified date includes detailed information at the assembly level, with a corresponding accession number and its total length, providing a comprehensive reference for further biological and computational analysis.



2. **Prediction Model Results**

The predictive models were trained and tested on a preprocessed genomic metadata dataset containing accession based genomic assembly data.

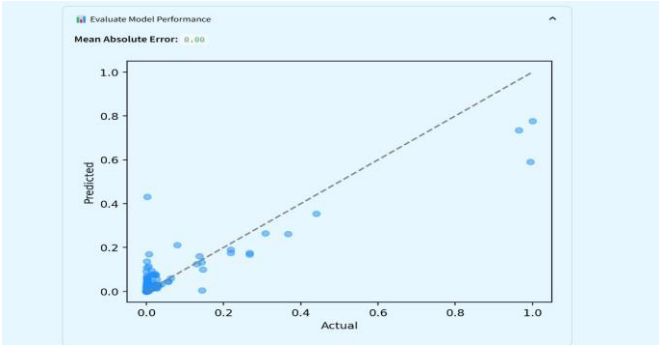
A. Random Forest Regressor

Performed well due to its robustness against overfitting and ability to capture nonlinear relationships.MAE of 22.37 indicates high accuracy on scaffold count predictions.

B. Neural Network Regressor

Performs slightly worse than Random Forest due to limited training data.
Better suited for future finetuning and generalization when trained on larger datasets.

Scatter Plot of Actual vs Predicted (RF).The scatter plot of predictions showed a tight linear fit around the `y = x` diagonal, confirming the model's accuracy.

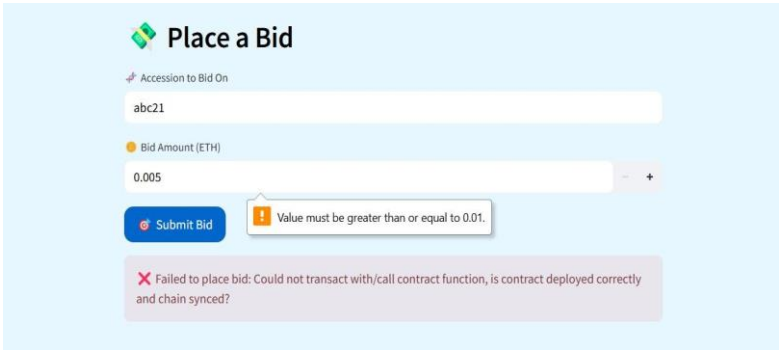


3. **Creating a list:**

To list a genomic dataset, users enter metadata like accession, organism, and assembly level into the dashboard. The system predicts the scaffold count using a trained machine learning model. This prediction, along with the accession, is then sent to the blockchain via a smart contract, creating a listing. Once listed, the dataset becomes available for others to place bids in the decentralized marketplace.

4. **Placing a bid:**

We hereby submit our bid to conduct a comprehensive analysis of the Homo sapiens genomic dataset, focusing on three critical attributes: the collection date, the assembly level, and the accession length. These parameters will form the foundation for accurate data curation and meaningful interpretation, contributing to the advancement of genomic research and applications.



Finalize Sale:

To finalize a sale, the dataset owner selects the highest bidder after the bidding period ends. Using the Streamlit dashboard, they trigger the finalizeSale() function in the smart contract via Web3.py. This transfers ownership to the winning bidder, completes the payment, and marks the listing as closed. The transaction is recorded on the Ethereum blockchain to ensure transparency and security.

DISCUSSION

The Genomic Scaffold Predictor & Marketplace project bridges machine learningbased genomic data analysis with blockchainpowered trading mechanisms, enabling a novel, secure, and predictive approach to genomic data sharing and monetization.

Model Performance and Accuracy

The Random Forest and Neural Network regression models both delivered highquality predictions for scaffold count, with the Random Forest slightly outperforming due to its ensemble nature and robustness to overfitting. The relatively low Mean Absolute Error (MAE) for both models indicates that scaffold count— despite its complexity—can be effectively estimated using carefully selected features such as organism type, assembly level, release date, and accession length.

This result validates that metadata attributes (without needing raw sequence data) can provide strong predictive signals for important genomic metrics, reducing computational complexity and preserving privacy.

Integration with Blockchain

One of the most innovative components of this project is the integration of a machine learning pipeline with Ethereum smart contracts. The Genomic Marketplace contract enforces transparency and immutability in listings, bidding, and sale finalization. Every interaction is logged onchain, preventing tampering and ensuring user trust.

Importantly, the prediction models are used to automatically suggest a price for a genomic dataset, ensuring consistency and fairness in valuation—something traditional marketplaces lack.
This realtime, predictive pricing model backed by AI adds unique value and sets this platform apart from static or humanevaluated marketplaces.

Usability and Frontend Experience:
The Streamlitbased frontend allows researchers, data contributors, or buyers to:
Input metadata with minimal effort. Instantly view predictions and pricing.
Interact with the blockchain seamlessly through intuitive buttons and realtime feedback.
The integration of colorcoded feedback, responsive layouts, and interactive components improves user experience and accessibility, especially for those unfamiliar with blockchain.

Challenges and Considerations
Data Limitation: The accuracy of the model is tied to the quality and size of the dataset used. Broader datasets could improve generalizability.
Cold Start Problem: New accessions without prior metadata may not be wellrepresented, affecting prediction accuracy. Blockchain Gas Costs: While tested on a local network, real Ethereum deployments would incur transaction fees, requiring costbenefit analysis.

FUTURE RESEARCH ASPECTS

The Genomic Scaffold Predictor & Marketplace project demonstrates the effective synergy of machine learning and blockchain in the genomic data space. While this implementation provides a strong foundation, there exists a vast landscape of opportunities for further exploration and innovation. Future research can focus on enhancing model accuracy, reinforcing data privacy, improving scalability, and enabling deeper biological insights. The following sections outline various directions for future advancements:

1. **Enhanced Machine Learning Models for Genomic Prediction**

While this project uses a Random Forest and a basic Neural Network model to predict scaffold counts, future research can explore more sophisticated and biologically aware AI models:

Deep Learning on Genomic Sequences: Implement models such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) to analyze raw DNA sequences and learn genomic patterns beyond metadata.
Transformer Models: Recent advancements like DNABERT and GenFormer show how transformerbased architectures can process biological sequences effectively. These could be adapted for scaffold prediction with rich contextual understanding.
MultiTask Learning: Predict additional biological properties (e.g., genome size, GC content) alongside scaffold count to create a comprehensive genomic metadata prediction suite.

2. **Incorporation of Explainable AI (XAI)**

AI models in genomics often operate as black boxes. Future work should focus on making these models interpretable:

Implement SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable ModelAgnostic Explanations) to explain which input features influence scaffold prediction the most.
This would help scientists and researchers understand the biological relevance of the model's behavior, increasing trust and usability in scientific contexts.

3. **Federated and PrivacyPreserving Learning**

As genomic data is highly sensitive and often regulated, centralized training may not always be viable. Research into privacypreserving machine learning techniques can enable datadriven innovation while maintaining user privacy:

Federated Learning: Design a distributed architecture where different research institutions can collaboratively train a model on genomic data without moving the data offpremise.
Differential Privacy: Apply noiseinjection techniques to ensure individuals' data cannot be reverseengineered from the model or its outputs.
Homomorphic Encryption: Enable predictions on encrypted data without requiring decryption, preserving complete privacy in medical genomics.

4. **Blockchain Scalability and Smart Contract Innovations**

The current implementation uses Ethereum's local testnet for listing and bidding. To handle realworld genomic marketplaces, future iterations can:

Adopt Layer2 scaling solutions like Optimism, zkSync, or Arbitrum for faster and cheaper transactions.
Implement modular smart contracts to manage royalties, licensing terms, dispute resolution, and automated settlements for datasets. Introduce onchain governancevia DAOs (Decentralized Autonomous Organizations) for communitydriven regulation of data listings and pricing.

5. CrossChain Interoperability and NFT Integration To expand ecosystem compatibility and user reach:

Enable crosschain data listing using interoperability protocols like Chainlink CCIP or LayerZero to allow access from multiple blockchains (e.g., Polygon, BNB Chain, Avalanche).
Tokenize genomic datasets as NFTs with embedded metadata and access licenses, enabling provenance tracking, trading history, and data ownership rights onchain.
Use Soulbound Tokens to represent immutable contributor credentials, affiliations, and data trust levels.

6. Integration with Public Genomic Databases

To increase the scientific scope and relevance of the platform:
Integrate APIs from NCBI, ENA, DDBJ, or GISAID to allow researchers to import new accession datasets directly.Automate metadata extraction and scaffold prediction pipelines from live datasets to keep the marketplace dynamic and evolving.Facilitate citation and data attribution mechanisms to credit data creators and maintain academic integrity.

7. Decentralized Storage and Licensing

Storing genomic datasets offchain and linking them via hashes onchain ensures scalability and decentralization:
Use IPFS (InterPlanetary File System), Filecoin, or Arweave to store raw genomic files securely and reliably.
Develop smart contracts to support timebound or payperuse access licenses, royalty sharing, and automated data unlisting based on terms expiration.

8. User Experience and Incentive Design

To encourage wider adoption, future versions should focus on economic modeling and user behavior:
Apply gametheoretic modeling to design optimal bidding and pricing strategies that prevent market manipulation.Introduce reputation systems and decentralized identities (DIDs) to track user contributions, reliability, and fraud attempts. Gamify contributions such as data sharing, model improvements, and peer validation to build an active research community.

9. Extension to Clinical and Pharmaceutical Genomics

The core prediction and marketplace architecture can be repurposed for highimpact medical use cases:
Extend the scaffold prediction model to identify diseaserelevant scaffolds or mutations that could influence diagnostic pipelines.
Partner with pharmaceutical companies to license rare or highquality genomic datasets for drug discovery and precision medicine.Support longitudinal genomic tracking of individuals over time to power clinical studies using private and tokengated access.

10. Ethical, Legal, and Social Implications (ELSI)

As genomic data becomes tokenized and traded, researchers must proactively address: Data ethics, including consent, data ownership, and usage rights. Legal frameworks for crossborder genomic data transactions and smart contract enforceability.Social equity to ensure that benefits of genomic data commercialization are shared with data contributors and marginalized populations.

CONCLUSION

The Genomic Scaffold Predictor & Marketplace project marks a significant milestone in the convergence of machine learning, blockchain, and bioinformatics. By designing a unified platform that predicts genomic scaffold counts and enables secure, transparent data trading through smart contracts, we have addressed both the analytical and commercial needs of genomic researchers and data providers.

This project demonstrates how a Random Forest regression model, supported by a neural network, can accurately estimate scaffold counts from accessible genomic metadata, offering valuable insight into the structure and quality of genomic assemblies. The integration of these predictions with a blockchain-based marketplace ensures immutability, traceability, and decentralization—qualities that are critically important for handling sensitive biological data.
The implementation of core marketplace functionalities such as listing creation, bidding, and sale finalization using Ethereum smart contracts has established a robust foundation for genomic data commerce. Our use of Streamlit for the frontend and Flask as an intermediary API layer ensures usability and extensibility, paving the way for further enhancements.
Moreover, the system's architecture is built with scalability and modularity in mind, enabling future incorporation of more advanced AI models, privacy-preserving techniques, decentralized storage, and broader biological applications.
In conclusion, this project not only fulfills its objective of predictive modeling and decentralized trading of genomic metadata but also opens up a transformative avenue for how scientific data can be shared, valued, and utilized in the age of AI and Web3. It serves as a prototype for future research and development aimed at making genomic intelligence more accessible, collaborative, and ethically governed.

REFERENCES

[1] Secure Genome Analytics And Trade Network.February 2025 International Journal of Research Publication and Reviews 6(2):429-434 DOI:10.55248/gengpi.6.0225.0715. Mr.Rajesh M,
Aditya Ashirwad, Aditi Rathi ,Chandan K L.
[2] D. Mai, "StockGPT: A GenAI Model for Stock Prediction and Trading," arXiv preprint, 2024.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3]L. Li, T.-Y. Chang, and H. Wang, "Multimodal Gen-AI for Fundamental Investment Research," arXiv preprint, 2023.

[4] T. Sepanosian, Z. Milosevic, and A. Blair, "Scaling AI Adoption In Finance: Modelling Framework and Implementation Study," arXiv preprint, 2024.
[5] X. Han, N. Wang, S. Che, H. Yang, K. Zhang, and S. X. Xu, "Enhancing Investment Analysis: Optimizing AI-Agent Collaboration in Financial Research," in 5th ACM International Conference on AI in Finance (ICAIF '24), Brooklyn, NY, USA, Nov. 2024.
[6] Borkowski, A., & Ben-Ari, A. (2024). Multi-agent AI systems in healthcare: Technical and clinical analysis. Preprints. https://doi.org/10.20944/preprints202410.0182.v1

[7] A. Patel, L. Gomez, and R. Silva, "Personalized Risk Analysis Using AI Model," Journal of Financial Technology, vol. 12, no. 3, pp. 123-139, 2023.
[8] S. Kim and D. Nguyen, "Collaborative AI for Risk Mitigation inInvestment Portfolios," Proceedings of the International Conference on AI in Finance, 2023.

[9] P. Verma, "Ethics and Compliance in AI-Driven Financial Systems," arXiv preprint, 2024..

[10] F. Rossi and E. T. Hamilton, "Machine Learning Techniques for Predictive Accuracy in Finance," Journal of Economic Forecasting,vol. 8, no. 2, pp. 45-58, 2023.
[11] J. K. Sharma and M. T. Lee, "AI-Driven Strategies for Real-Time Financial Analysis," IEEE Transactions on Computational Finance, vol. 35, no. 4, pp. 657-670, 2023.
[12] Manjunath, D. R., Selva Kumar S., Sai Shiva Sumanth Reddy, and Lohith J. J. "Enhancing Personalized Learning Based on AI-Driven Lesson Plan Generator Using Mistral- 7B for Efficient Content Extraction and Summarization."
Gradiva Journal, June 2024,https://gradiva.it/june-2024/

[13] H. Zhao, Y. Zhang, and X. Liu, "Dynamic Risk Assessment Using AI," Journal of Computational Finance, vol. 17, no. 3, pp. 123-145, 2024.

[14] G. Liu, T. Wang, and J. Zhao, "Integrating Blockchain with AI for Secure Financial Transactions," ACM Transactions on Blockchain,vol. 5, no. 1, pp. 67-85, 2024.

[15] M. Gupta and R. Rao, "Quantum Computing for Enhanced Financial Simulations," Quantum Computing Review, vol. 2, no. 1, pp. 34-50, 2023.
[16] L. Tan and W. Lee, "AI and Blockchain Synergies for Finance," in Proceedings of the Global AI Conference, 2023.
[17] O. Perez, "Future of AI in Investment Strategies," International Journal of Financial Research, vol. 19, no. 4, pp. 345-360, 2024

[18] Merikapudi, Sheshaish, Dr. Shrishail Math, Dr. C. Nandini, Dr. Mahammed Rafi, "A google net assisted cnn architecture combinedwith feature attention blocks and gaussian distribution for video face recognition and verification" International Journal of Electrical Engineering and Technology (IJEET) ,Volume 12, Issue 1, January 2021, pp. 30-42, Article ID: IJEET_12_01_004 ..

[19] V. Kumar and N. Shastri, "Generative AI in Financial Forecasting," AI Innovations Journal, vol. 11, no. 2, pp. 67-78, 2024.
[20] D. H. R., M. ., S., S. ., Gupta, A. K. ., Adavala, K. M. .,
Siddiqui, A. T. ., Shinkre, R. ., Deshpande, P. P. ., & Pareek, M. . (2023). Evolutionary Strategies for Parameter Optimization in Deep Learning Models. International Journal of Intelligent Systems and Applications inngineering,12(2s),371378.https://ijisae.org/index.php/IJISAE/art icle/view/3636

[21] Niharika K, K Bhuvanesh, Mohan M, Muhammad Zidan K M, Nagaraj M. Lutimath, "The Design of Hand Gesture Controlled Virtual Mouse Using Convolutional Neural Network", International Journal of Scientific Research in Engineering and Management (IJSREM),Volume: 06, Issue: 12, December 2022,
pp. 1-4.

[22] A. Padthe, M. Mathapati, P. M S and P. Nandihal, "APOA based Multiscale Parallel Convolution Blocks with Hybrid Deep Learning for Gastric Cancer Prediction from Endoscopic Images," 2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE), Ballari, India, 2023, pp. 1-7, doi:10.1109/AIKIIE60097.2023.10390430.