AI Powered Weather Insurance System on a Blockchain

1st Ben Taylor
School of Computing
Newcastle University
Newcastle, UK
b.taylor9@newcastle.ac.uk

2nd Eliot Young
School of Computing
Newcastle University
Newcastle, UK
e.j.young2@newcastle.ac.uk

3rd Nikodem Jandernal
School of Computing
Newcastle University
Newcastle, UK
n.jandernal2@newcastle.ac.uk

4th Jeny Wightman
School of Computing
Newcastle University
Newcastle, UK
j.wightman3@newcastle.ac.uk

5th Lucy Garden
School of Computing
Newcastle University
Newcastle, UK
l.garden2@newcastle.ac.uk

6th Sakai Newman
School of Computing
Newcastle University
Newcastle, UK
s.newman2@newcastle.ac.uk

7th Wesley Smith
School of Computing
Newcastle University
Newcastle, UK
w.smith7@newcastle.ac.uk

8th Cosmin Irimescu
School of Computing
Newcastle University
Newcastle, UK
c.irimescu2@newcastle.ac.uk

I. Literature Review

The increasing number of extreme weather events has highlighted the need for efficient, transparent and fair insurance systems. We propose an automated decentralised weather-based insurance system using smart contracts, sensor data and CNN-based image damage assessments. In this literature review, we explore pre-existing sources relating to these ideas.

Smart contracts are a key component of our proposed system. Hawk [1] suggests a privacy-focused approach using cryptographic techniques to hide financial transactions from the public blockchain. This allows developers to write contracts without needing much cryptographic expertise, addressing key privacy concerns while maintaining the decentralisation and security benefits of blockchain technology. Alongside this in Zether: towards privacy in a smart contract world [2] introduces a confidential payment mechanism utilising encrypted account balances and cryptographic proofs for deposits, transfers and withdrawals. It aims to ensure confidential transactions while preventing accidental loss of funds, showing smart contacts can provide privacy without losing functionality or limiting performance overhead.

Our system will use deep learning for image-based damage assessment. ImageNet Classification [3] highlights how deep CNN architectures can extract high-level image features automatically. This can be used to identify cracks, collapsed structures, or missing elements in damaged buildings, which would assist in insurance claim validation. CNN(Convolutional Neural Network)-based models often inherit biases from training data, as discussed on ConvNets [4], which introduces a technique to track learnt biases over time.

The European Union Regulations on algorithm design [5] explore how insurers must provide a way for human review if a decision is fully automated, meaning a person must be available to reassess the algorithm's decision upon request. Concrete problems in AI safety [6] identifies key safety concerns in AI systems, specifically regarding the objective of its function, its evaluation and unintended behaviours that may arise during learning. It covers issues such as reward hacking and scalable oversight and introduces key mitigation strategies against said concerns.

Fraudulent claims, system abuse and bot spamming pose significant risks to automated insurance platforms. Statistical fraud detection [7] demonstrates how using natural networks and machine learning models to assign suspicion scores to claims based on statistical patterns can aid in fraud detection. A claim is flagged if it deviates significantly from the norm, such as being unusually expensive. Botminer: Clustering analysis [8] outlines how botnets can be detected by clustering communication traffic and malicious activity patterns and then correlating these clusters to identify bots. It explores clustering for anomaly detection and cross-correlation and results show positive detection of real-world botnets, with a low false positive rate. Implementing similar strategies can enhance fraud detection and system integrity.

Our system must also defend against adversarial attacks. Certified Defences for Data Poisoning Attacks [9] presents defence frameworks against data poisoning, identifying outlier detection as the most effective strategy. Similarly, Intriguing Properties of Neural Networks [10] explores two properties of deep neural networks: their feature representations are distributed rather than isolated to specific neurons and are highly vulnerable to imperceptible adversarial perturbations. These findings raise concerns about robustness and interpretability, underlining the need for improved security and generalisation in neural network design.

II. Threat Model

A. What is our Project?

We will create an automated decentralised weather-based insurance system using sensor data and CNN-based image damage assessment. Smart contracts will manage automatic payments, and if fraud is detected, claims will be reviewed manually by our employees. Users will be able to request a manual claim review. Both larger claims and some random ones will require manual reviewing.

B. Who is our Adversary?

Our main adversaries are insurance fraudsters trying to 'trick' our system into incorrectly paying out their insurance claims. Cybercriminals will try to exploit vulnerabilities to either steal personal data and money or take down the system. Similarly, rival insurance companies may try to disrupt our system for their benefit.

C. What are some Potential Threats?

Some potential threats include the manipulation of weather sensors and AI-generated images designed to fool the CNN. Additionally, hidden patterns could be added to the images. The main goal of these types of threats is to produce false positives for a successful claim. Other threats include vulnerabilities within the smart contracts to alter payments. Competitors may try to either perform denial of service attacks or reverse engineer the system

D. What are the Attackers' Capabilities?

Attackers can intercept and manipulate sensor data and generate fake or doctored images. Also, we anticipate attackers may read any unencrypted data or at least make inferences from poorly encrypted data, such as working out a user's location due to their proximity to sensors and the images that capture property damage. Attackers may try to use malicious inputs within any publicly accessible function.

III. Security Policy

A. Data integrity and authenticity

- Encrypt all data using non-malleable encryption plus timestamps or nonces.
- Digital signatures for images to ensure authenticity and prevent tampering.
- Voting system for nearby sensors to identify and remove anomalies caused by faulty or malicious sensors.
- Generalise addresses to areas to ensure adversaries cannot find someone's address through vulnerabilities.
- Limit data exposure to minimum needed ensuring that only necessary data is accessed for claim processing.

B. Fraud detection and prevention

- Third-party verification by employees when deemed necessary.
- Checking for fraudulent images (reverse image search).
- Audit logging for insurance claims, and staff actions.

C. Security of smart contracts

- Implement formal verification (e.g. Dafny) to ensure correctness, close loopholes in the code logic, and securely program each publicly accessible function with proper input validation, constraints, and preconditions.
- Time locking to prevent large rapid withdraws often linked with fraud.
- Implement an RBAC(Role Based Access Control) policy that restricts smart contract functions based on predefined roles.

D. Overall system resilience and defence against specific adversarial capabilities

- Implement CAPTCHA verification to defend against bots.
- IP tracking to detect suspicious activities and identification of potential cyber criminals.
- Rate limiting to control network traffic by limiting the number of requests made to a server by a user which helps prevent DDOS(Distributed Denial of Service) attacks.
- RBAC - Admin account, Reviewer, Customer.

E. Regulatory compliance and user trust

- Ensure compliance with GDPR(General Data Protection Regulation) data protection laws for users.
- Maintain transparency in claim processing to ensure customer trust and belief in the system.
- Provide explainability for CNN models to improve AI model trust.
- Claim resolution system for users to request a review.

F. Image analysis

- Defence against adversarial attacks such as Photoshop.
- Reverse lookups and metadata checks.
- Perceptual hash algorithms and digital signatures.

IV. High level design

A. System Overview

The system is an automated, decentralised weather-based insurance system, allowing a platform for autonomous insurance claim payouts. The project combines weather sensor data and CNN-based image assessment to make decisions on payouts, which are managed through smart contacts. Robust security measures, such as digital signatures, encryption and anomaly detection are to be implemented to mitigate fraud by marking suspicious claims for manual review. The frontend of the project will allow users or staff to login to their accounts, where they can submit claims or review claims respectively.

Figure 1 describes the technologies and architectures used for development of the project and how they integrate.

Figure 2 provides a high-level overview of how the system is structured on a conceptual level, which delves into the specifics of how the system operates.

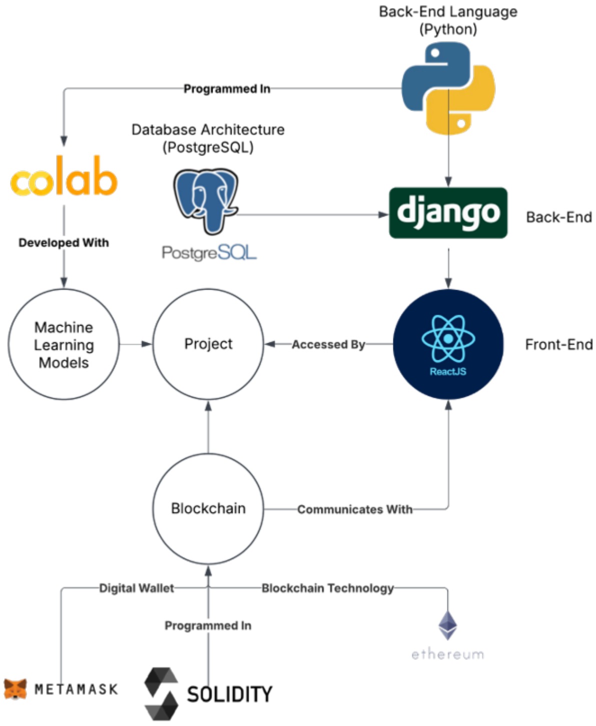

Fig. 1. System Architecture Diagram.

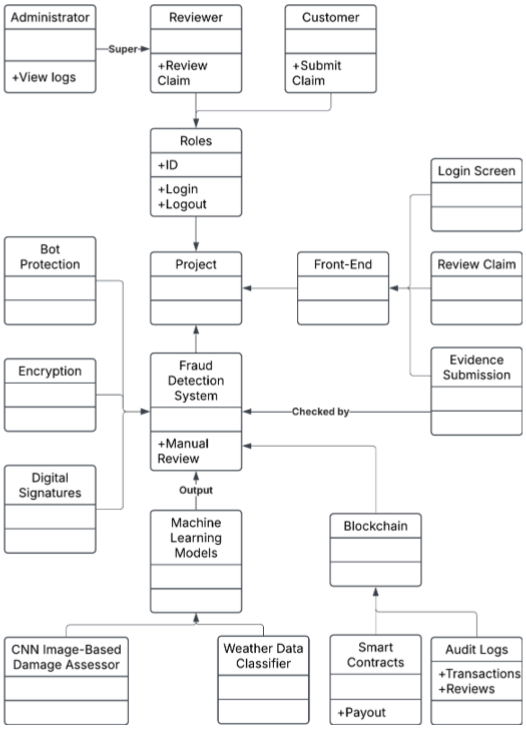

Fig. 2. System Overview UML Diagram.

B. Data Collection and Processing Pipeline

1) Data Acquisition: The project will require a large amount of data, both weather-based and image-based. For the weather data, many publicly available datasets exist which document weather in each area. Images of both damaged houses and non-damaged houses will be required in order to train a model that can distinguish between these two states. These will be gathered from publicly available datasets online.

2) Data Preprocessing: Data preprocessing will be necessary on our collected data, in order to facilitate the model's training process. Both weather and image data will need to be normalised to a consistent range across all input fields. In addition, outliers in the data will need to be identified and removed. This is necessary to defend against data poisoning attacks [9] as well as dealing with malfunctioning weather sensors and other benign outliers.

3) Data Flow & Integration: The information in the system is structured around discrete points of integration. User uploaded images, as well as sensor data are sent securely through encrypted connections. The data is initially validated as well as pre-processed at that point to preserve the integrity before sending it to the machine learning module. In the module, sensor data is categorised, with a neural network charged with determining damage from images. The results are securely transferred via our REST APIs over to the layer of blockchain via Ether.js, in which smart contracts carry out functions like validation of claims, as well as payment processing and produce an audit trail for documentation of transactions.

C. Blockchain and Smart Contract Layer

Smart contracts will facilitate the automatic payout of an insurance policy to a claimant upon the validation of a claim. Claims will be validated by the ML models or through a review conducted by an employee. The system deploys multiple smart contracts on a local Ethereum blockchain, logging all interactions and transactions.

1) Smart Contract Design: Four smart contracts are required for automatic payments: an insurance payment contract, an insurance policy contract, a claim submission and validation contract, and an automated payment contract. The insurance payment contract allows users to pay into their insurance policy. The insurance policy contract stores policyholder details, coverage information, and records of premium payments. The claim submission and validation contract manages claim metadata and interacts with ML models to validate claims. The automated payout contract is responsible for releasing funds to a claimant's digital wallet using escrow accounts.

2) Security, Transparency and Auditability: To ensure correctness and transparency, the system incorporates formal verification through Dafny to verify smart contract correctness. Additionally, the smart contract design will include timelocking mechanisms to safeguard against fraudulent transactions. Smart contract functions will also be protected using RBAC to prevent adversaries from unauthorised access and modifications.

All smart contract interactions and transactions will be recorded on the local Ethereum blockchain instance, allowing for real-time auditing. Auditing ensures non-repudiation of origin for sending funds and non-repudiation of receipt for receiving funds from a claim payout.

D. Security and Resilience Measures

1) Security Controls: Data confidentiality is maintained through end-to-end encryption for sensor readings and user claims [2]. Data integrity is guarded through various verification methods. Digital signatures and hashing validate images to prevent tampering and duplications [8]. A voting-based anomaly detection system ensures sensor data authenticity through cross-verifying weather readings from multiple sources. Audit logging on the blockchain increases transparency by recording insurance claims, system interactions and smart contracts ensuring accountability.

Access control mechanisms prevent unauthorised system interactions. RBAC defines strict rules for different user roles. Administers oversee security and fraud detection, and reviewers validate AI generated classifications and flag claims. Multi-factor authentication reduces credential-based attacks, while smart contracts prevent unauthorised transaction execution [1].

Fraud detection integrates machine learning models to identify statistical anomalies in claim submissions [7]. Reverse image search detects duplicate claim submissions, and time-locking mechanisms to delay large payouts allowing additional verification.

2) Threat Mitigation: Our system faces threats from fraudulent claims, adversarial AI attacks and smart contract vulnerabilities. Fraudulent claims are mitigated through anomaly detection on sensor data and adversarial-resistance CNN training to detect AI-generated forgeries [4]. The manual claim review mechanism is implemented for high sum and randomly selected claims to prevent exploitation.

Adversarial AI attacks involve attempts to manipulate the neural networks into incorrectly classifying fraudulent claims, this can be addressed via adversarial training, which improves CNN robustness. While metadata validation ensures image authenticity through GPS tags and timestamps [10]. Finally distributed denial-of-service attacks are countered through rate limiting, CAPTCHA verification, and IP tracking to detect and block suspicious activity.

E. Interfaces and Integration

Non-Users will be allowed to make an account or log in to an existing account.

The UI will allow users to make an insurance claim by uploading images and filling in claim details, tracking the state of claims, requesting a review and paying into insurance policies as well as cancelling the policy.

The reviewer interface will allow the reviewer to view flagged claims, validate AI classifications, and comment on disputed claims.

The admin interface manages user's roles, permissions, monitors fraudulent activity and claims. Also to audit smart contract transactions, view machine learning model performance and audit logs.

F. Tools, Technologies, and Project Management

For our project, the front-end will be developed using ReactJS to create a responsive user interface. Data will be stored in a PostgreSQL database, which will integrate well with our Django back-end, where we will use Python as our primary language. Smart contracts will be written in Solidity and deployed on the Ethereum blockchain, with MetaMask serving as the digital wallet for storing cryptocurrency, paying gas fees, and handling transactions. Tools like HardHat will be used for deploying a local blockchain instance, and JavaScript will interact with the blockchain on the front-end.

To train the CNNs (Convolutional Neural Networks), Google Colab will be used, which provides us with the computational power as well as ease of use. Code management will be done through GitHub, with team communications as well as regular updates on progress done on Microsoft Teams.

G. Testing & Evaluation

1) Testing Strategy: When testing the system, unit tests are used to verify that the individual components of the application work as expected. Integration tests are then used to check that these individual components work together as intended when working as one unit, then followed by system tests to test the overall functionality of the application. Finally, security tests will be added to test the effectiveness of the implemented security features.

2) Evaluation Metrics: For evaluation, there should be a focus on evaluating weather data accuracy, the performance of the CNNs, smart contract performance and overall user experience. To start off, we would measure the accuracy of the weather sensor data by comparing the sensor readings with known weather reports and then determine the percentage accuracy of the data. We would then look at recording the precision, recall and F1 scores to ensure that the CNN classifies the damage accurately without much error. Additionally, when evaluating the performance of the smart contracts, we would look at measuring gas efficiency to assess the cost-effectiveness of their deployment, followed by assessing the execution time to see how long it takes for a claim to be processed on the blockchain. We then would evaluate user experience by setting up a review system to measure user satisfaction.

H. GDPR response policy

1) Incident Detection & Reporting: Data breaches will be identified based upon regular audits and logging performed throughout the system. Our data controller will be responsible for reporting incidents within 72 hours of discovery [11]. This will be to both authorities and customers.

2) Containment & Mitigation: Encrypted data minimises the impact of damages by preventing attackers from accessing plaintext data. An internal staff investigation and revocation of accounts suspected to be hostile will be conducted. A password reset will be enforced for impacted users if necessary.

3) Breach Notification Procedures: We will notify users within the web application as necessary. The authorities will be informed with all the required information unless an exception occurs, such as when the impacted data is encrypted, or risk mitigation steps have already been implemented.

4) Forensic Analysis and Root Cause Identification: As part of our forensic investigation into incidents, we will collect relevant information such as logs, database records and blockchain transactions. Then we will review the data to discover the entry point used by the attacker to gain access to the system. Once the attack path has been discovered we will review the impact this has had on the confidentiality and integrity of data and respond based on the results.

5) Post Breach Measures: Containment and damage control will be conducted to identify and isolate affected systems. Compromised credentials will be reset or revoked, vulnerabilities patched, and any malware removed. A breach investigation and risk assessment will follow, along with notification to the relevant authority if required. An audit log of all data breaches will be maintained, even if reporting is not necessary.

I. Documentation and Future Improvements

1) Scalability Considerations: To ensure scalability, we will optimise gas efficiency within smart contracts and strictly enforce input validation to reduce exploitability. Dafny will be used to ensure this. Various penetration testing will be performed on all aspects of the project. Formal verification should reduce the number of exploitable sections of code which will reduce memory usage. Rate limiting will ensure services cannot be disrupted.

2) Future Improvements: To further enhance this project, we would utilise data from publicly available cameras to verify claims if available in the area. Additionally, opening the insurance system to a wider variety of markets such as car insurance would increase the number of potential users. Given the cars contained enough data sensors, this could also be combined with road traffic cameras and eyewitness testimony.

References

[1] Kosba, A., Miller, A., Shi, E., Wen, Z. and Papamanthou, C. (2016). Hawk: the Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts. 2016 IEEE Symposium on Security and Privacy (SP), pp.839–858. doi:https://doi.org/10.1109/sp.2016.55.

[2] Benedikt Bünz, S. Agrawal, M. Zamani, and D. Boneh, “Zether: Towards Privacy in a Smart Contract World,” Cryptology ePrint Archive, 2019. https://eprint.iacr.org/2019/191

[3] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2017). ImageNet Classification with Deep Convolutional Neural Networks. Communications of the ACM, 60(6), pp.84–90. doi: https://doi.org/10.1145/3065386.

[4] Stock, P. and Cisse, M. (n.d.). ConvNets and ImageNet beyond Accuracy: Understanding Mistakes and Uncovering Biases. Available at: https://arxiv.org/abs/1711.11443

[5] B. Goodman and S. Flaxman, “European Union Regulations on Algorithmic Decision-Making and a ‘Right to Explanation,’” AI Magazine, vol. 38, no. 3, pp. 50–57, Oct. 2017, doi: https://doi.org/10.1609/aimag.v38i3.2741.

[6] Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., Dan, O. and Google Brain, M. (2016). Concrete Problems in AI Safety. [online] Available at: https://arxiv.org/pdf/1606.06565.

[7] Bolton, R.J. and Hand, D.J. (2002). Statistical Fraud Detection: a Review. Statistical Science, 17(3). doi:https://doi.org/10.1214/ss/1042727940

[8] Gu, G., Perdisci, R., Zhang, J. and Lee, W. (n.d.). BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-Independent Botnet Detection. [online] Available at: https://www.usenix.org/legacy/event/sec08/tech/full_papers/gu/gu.pdf.

[9] Steinhardt, J., Pang, W., Koh and Liang, P. (n.d.). Certified Defenses for Data Poisoning Attacks. doi:https://arxiv.org/abs/1706.03691

[10] Szegedy, C., Zaremba, W., Ilya Sutskever, Bruna, J. and Fergus, R. (2013). Intriguing Properties of Neural Networks. [online] Researchgate. Available at : https://www.researchgate.net/publication/259440613_Intriguing_properties_of_neural_networks

[11] intersoft Consulting (2013). General Data Protection Regulation (GDPR) – Final text neatly arranged. [online] General Data Protection Regulation (GDPR). Available at: https://gdpr-info.eu/art-34-gdpr/.