Webmob Software Solutions
Engineering the Future

AI-DRIVEN INSURANCE CLAIM FRAUD DETECTION SYSTEM

TABLE OF CONTENT

Case Study

AI-DRIVEN INSURANCE CLAIM FRAUD DETECTION SYSTEM

Webmob Software Solutions

Bestech Business Tower, Suite No. A-829,
Sector - 66, Mohali, Punjab 160066

3030 K Street NW, Suite 102 Washington, DC 20007

0172-4045981

info@webmobinfo.ch

ABOUT WEBMOB

Webmob has emerged as a service delivery pioneer in this dynamic fintech industry, serving numerous laurelled clients in Europe and the Middle East. With AI/ML-powered, Cloud-native, and Blockchain in our stack, Webmob provides cutting-edge solutions to fulfil the customer's advanced and disruptive requirements.

Particularly for the FINTECH industry, Webmob offers unparalleled robust solutions in Trade Finance, Money Market, Fiduciary, Commercial Real Estate Loan Tokenization, and NFT marketplace on top Blockchains. As of today, Webmob is equipped with a fully equipped R&D lab, aka WikiDLT.com, and consulting certified professionals, especially to explore new possibilities for innovative Blockchain implementation.

OVERVIEW

Our platform uses cutting-edge voice analytics to detect false claims in the insurance sector. The innovative approach uses encoder-decoder architectures, specifically Generative Adversarial Networks (GANs). This advanced technology allows us to identify discrepancies and potential fraud more accurately and efficiently.

Initially, we found that embeddings generated directly from voice samples did not yield the desired results. We transformed these voice embeddings into images to overcome this, significantly enhancing our detection capabilities. By converting audio data into visual representations, our system can better analyse and identify patterns indicative of fraudulent activity.

Our platform's unique methodology improves the precision of fraud detection and streamlines the entire claims process. By integrating sophisticated voice analytics with GANs; we have ensured higher security and reliability for insurers. This technological advancement empowered us to offer a robust solution that mitigates risk and safeguards the interests of insurance companies and their clients.

Business Needs

- Reduce the annual loss from insurance claim fraud.
- Improve customer service by reducing compliance delays.
- Enhance fraud detection security.
- Increase accuracy in identifying fraudulent claims.
- Ensure data integrity and confidentiality.

Our Solutions

- Integrated third-party inventories for product diversity.
- Automated order processing and fulfillment.
- User-friendly platform for easy listing and purchasing.
- Detailed inventory search for quick part finding.
- Transparent transactions with partial supplier payments.

Benefits

- Reduce fraudulent claim payouts significantly.
- Speed up claim processing for better customer satisfaction.
- Increase accuracy in fraud detection.
- Ensure data security and privacy.
- Integrate seamlessly with existing insura technologies.

OUR SOLUTIONS

01 Voice Recognition

We implemented artificial intelligence and machine learning technologies to analyse the voice patterns of individuals making insurance claims. By examining the tonal quality and voice modulation, our system can detect inconsistencies and potential fraud, effectively identifying when a fraudster attempts to impersonate a legitimate claimant.

02 Sentiment Analysis

Our platform uses sentiment analysis to evaluate the emotional tone and sentiment in speech and text communications during the claims process. Additionally, by identifying unusual emotional cues or irregularities in the claimant's responses, we can flag suspicious activities and reduce the likelihood of fraudulent claims slipping through.

03 Voice to Image Conversion

To enhance fraud detection accuracy, we convert voice data into visual representations. These images capture the unique patterns of the claimant's voice. Our system then compares these visual patterns to detect anomalies or changes that may indicate fraud, providing a more reliable verification method.

04 Blockchain Integration

We incorporated R3 Corda blockchain technology to store and manage claim data securely. It ensures that all data is anonymised and stored in a tamper-proof environment, maintaining high data integrity and confidentiality standards. Moreover, it also allows for secure data sharing across the insurance ecosystem while preserving privacy.

05 Credibility Scoring

Our platform developed sophisticated algorithms to generate claim credibility and confidence scores. These scores are derived from the combined voice recognition, sentiment analysis, and voice-to-image conversion analysis. Claim handlers can use these scores to assess the reliability of each claim, making the decision-making process more efficient and accurate.

TECHNOLOGY

Blockchain - R3 Corda

We utilised R3 Corda for secure data storage and sharing, ensuring anonymity and integrity of claim data.

Python - Language

Our primary language for developing system components facilitates rapid development and maintenance.

Flask - API Creation

We have used Flask to create lightweight, flexible APIs for seamless communication between system modules.

Wave Library - Audio Processing

We have enabled the extraction of key voice characteristics for fraud detection from audio recordings through Wave library.

Numpy - Image Matching

We have facilitated efficient image matching and analysis for visual representations generated from voice data using the Numpy library.

Scikit - Voice to Image Conversion

We have converted voice embeddings into visual representations, enhancing fraud detection accuracy through novel pattern recognition.

Challenges

Earlier, insurance companies faced significant challenges in effectively detecting and preventing claim fraud. Traditional methods relied heavily on manual review processes, which were time-consuming, prone to human error, and often lacked accuracy. Moreover, fraudsters continually evolved tactics, making it increasingly difficult to identify fraudulent claims using conventional techniques alone. The need for advanced technological solutions meant that insurers struggled to keep pace with the sophisticated methods employed by fraudsters, resulting in substantial financial losses of billions annually in the US alone.

Additionally, existing systems often needed more integration capabilities, making it challenging for insurers to streamline processes and improve efficiency. This fragmented approach led to delays in claim processing, increased compliance burdens, and ultimately, diminished customer satisfaction. In this environment, there was a pressing need for a comprehensive and technologically advanced solution to address these challenges and empower insurance companies to combat fraud more effectively.

QA PROCESS

Our QA process involves a systematic approach encompassing various stages to thoroughly assess the trade finance platform's functionality, security, and user experience.

01 Test Planning
We defined the scope of testing, identified objectives, allocated resources, and developed a comprehensive test plan outlining our approach, timelines, and deliverables.

02 Requirement Analysis
We reviewed the requirements documentation to understand the expected behaviour of the trade finance platform and ensured our team accurately captured all functional and non-functional requirements.

03 Test Environment Setup
We established testing environments mirroring the production environment, installed the necessary software, configured databases, and ensured the availability of test data representing various trade finance scenarios.

04 Test Case Design
We developed detailed test scenarios and cases covering functional workflows, boundary conditions, error handling, and exception scenarios, prioritising based on criticality and risk.

05 Functional Testing
We executed test cases to verify the functionality of different modules and features, including various trade finance workflows, integration with external systems, and compliance with regulatory standards.

06 User Interface Testing
We evaluated the user interface for usability, accessibility, and responsiveness, ensuring consistency in design elements, layouts, and navigation across different screens.

07 Security Testing
We performed security assessments, testing authentication and authorisation mechanisms, data encryption, and secure communication protocols, and conducted penetration testing to assess resilience to security breaches.

08 Regression Testing
We reran previously executed test cases to ensure new changes did not introduce any regressions, automating regression test cases where feasible and validating backward compatibility.

09 Integration Testing
We tested data exchange mechanisms and validated data consistency and integrity across integrated systems, including file uploads, API calls, and message queues.

10 Documentation and Reporting
We maintained a detailed documentation of test cases, results, and defects, generated test reports summarising test coverage and provided stakeholders with regular updates on testing progress and identified issues.

11 User Acceptance Testing (UAT)
We collaborated with end-users and stakeholders to conduct UAT, obtaining feedback on the platform's functionality, usability, and performance, ensuring alignment with user expectations.

SECURITY TESTING OF THE PLATFORM

01 API Testing

Objective: Evaluate the functionality, reliability, security, and performance of APIs used in the platform.

Tools:

- **Postman:** Automated testing tool for API automation testing, enabling comprehensive testing of API endpoints and payloads.
- **SoapUI:** Another automated testing tool suitable for API testing, providing features for functional testing, load testing, and security testing.

02 Penetration Testing (PenTesting)

Objective: Identify and exploit vulnerabilities in the platform to assess its security posture.

Tools:

- **Burp Suite:** A comprehensive toolkit for web application security testing, including manual and automated vulnerability scanning, request interception, and exploitation of security flaws.
- **Metasploit:** A penetration testing framework offering various exploits and payloads for testing network and application security.

03 Patch Testing

Objective: Verify the effectiveness of security patches applied to the platform.

Process:

- Testing patches on a sandbox or staging environment ensures they do not introduce regressions or new vulnerabilities.
- Automated and manually tested critical functionalities affected by the patch to ensure they operated as expected.

04 Third-Party Testing

Objective: Gain independent verification and validation of the platform's security measures.

Process:

- Engaging external security firms or independent security researchers to conduct thorough security assessments, including penetration testing, code review, and vulnerability scanning.
- Utilising bug bounty programs to incentivise external security researchers to discover and responsibly disclose security vulnerabilities in the platform.

05 Source Code Testing

Objective: Evaluate the security of the platform's source code to identify and remediate vulnerabilities and ensure robust protection against potential threats.

Process:

- The source code testing process for the platform begins with configuring and integrating tools like SonarQube and Checkmarx into the development environment.

Tools:

- **SonarQube:** Analyzes the platform's source code for bugs, vulnerabilities, and code smells, providing insights into code quality and security.
- **Checkmarx:** A static application security testing (SAST) tool that identifies security vulnerabilities in the source code, helping developers remediate potential issues before deployment.

06 Network Testing

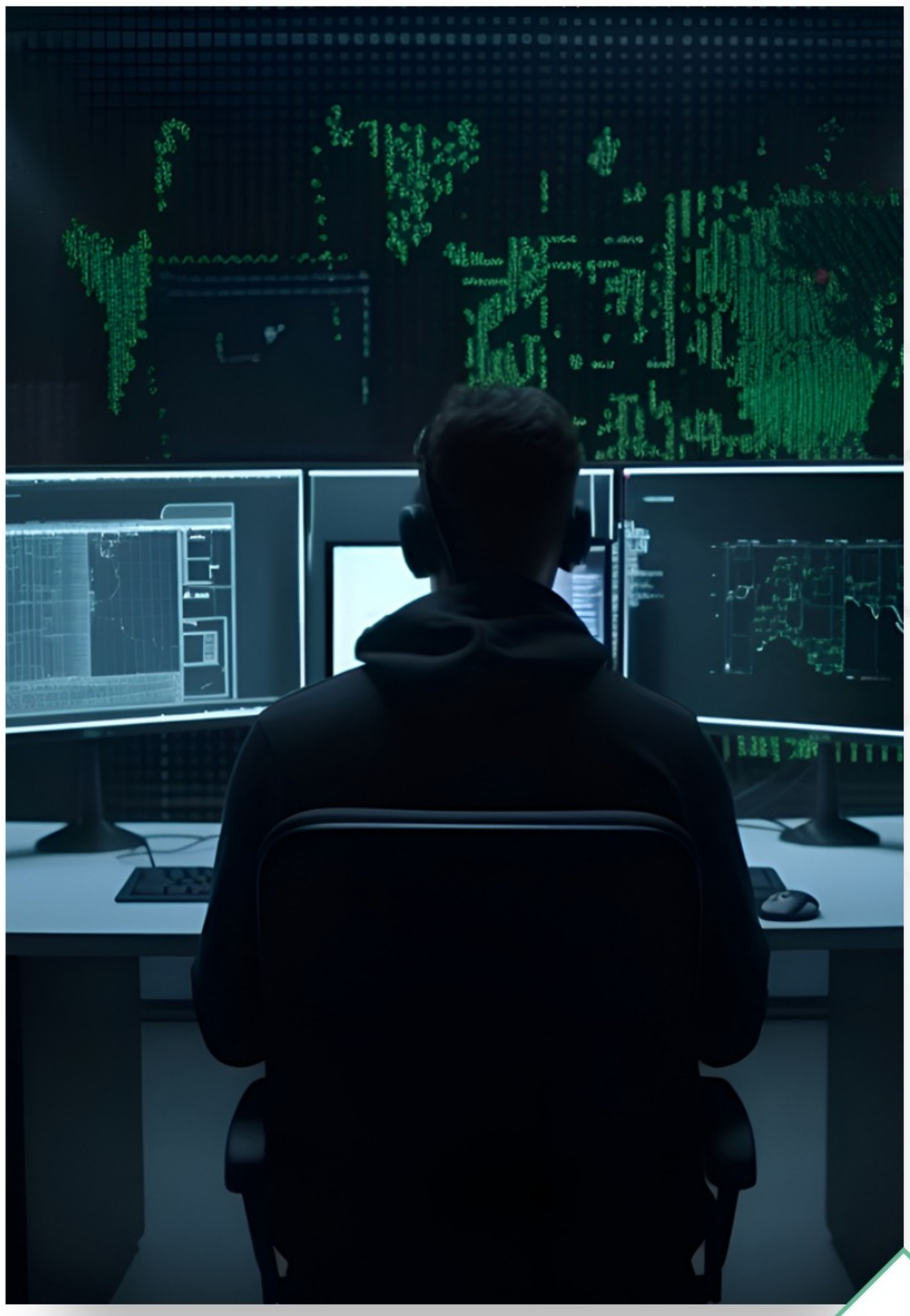
Objective: The primary objective of network testing is to assess the security and resilience of the platform's network infrastructure, ensuring protection against potential threats and vulnerabilities.

Process:

- Network testing begins by examining the network infrastructure's configuration and setup to identify any potential weaknesses or misconfigurations.
- Comprehensive scans are conducted using specialised tools to analyse server ports, configurations, versions, and subdomains within the network.

Tools:

- **Nessus:** A powerful scanning tool utilised for comprehensive network scans, providing detailed insights into potential security risks and vulnerabilities within the network infrastructure.
- **Nmap:** Another widely used scanning tool that enables thorough examination of network configurations and identifies potential security loopholes and weaknesses.



DEVELOPMENT PHASE

01 Requirement Gathering

Requirements were gathered through meetings and discussions to understand trade finance's functional and non-functional aspects.

02 System Design

Based on the gathered requirements, system architecture and design were finalised. It included defining the database schema, application modules, and integrations with external systems.

03 Coding

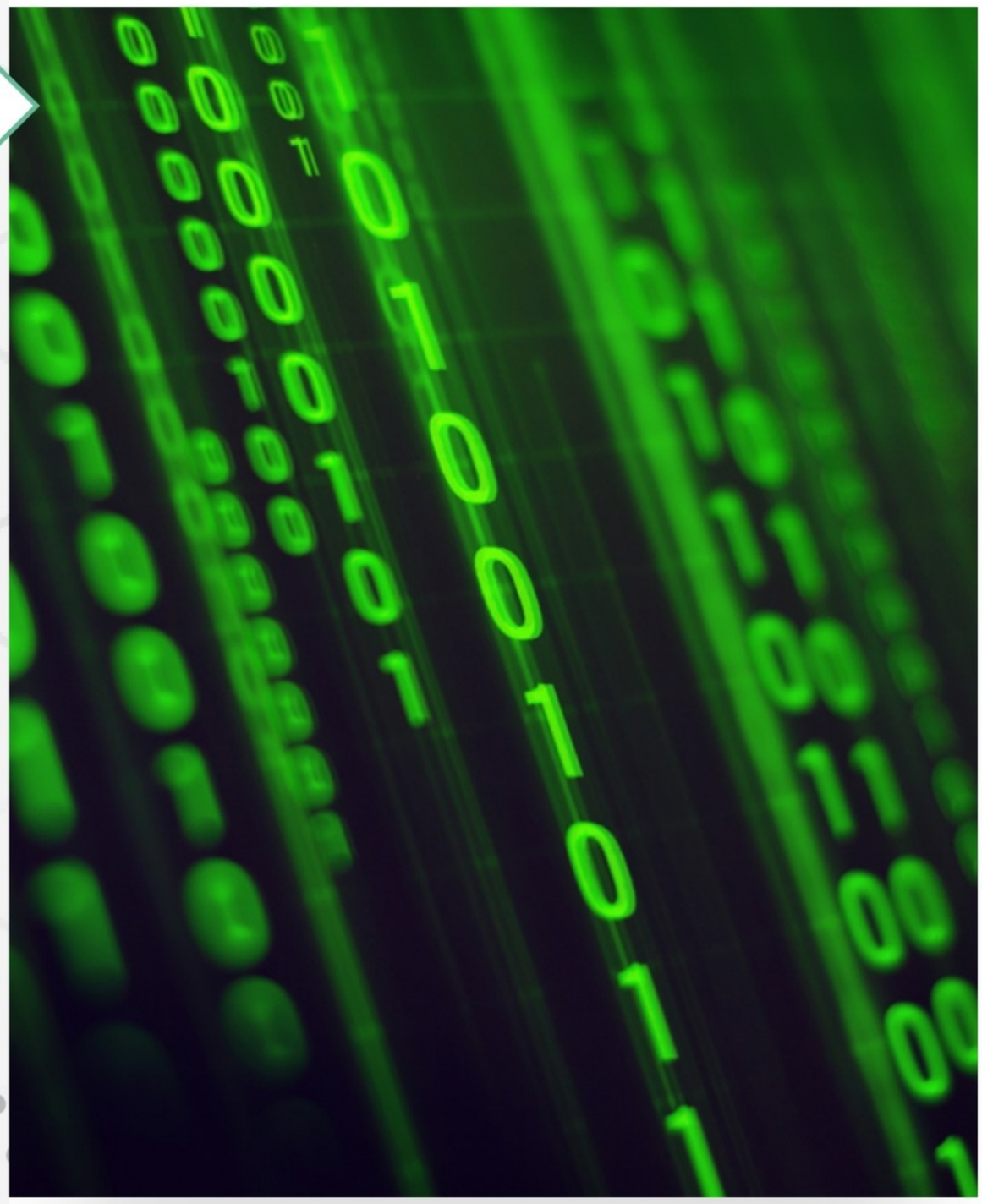
Our developers wrote code according to the design specifications using programming languages & frameworks suitable for the platform's requirements.

04 Quality Assurance

Our QA engineers conducted comprehensive testing of the platform, including source code, functional, security, and performance testing that helped us identify & resolve any defects or issues.

05 Review & Integration

The platform has undergone thorough code reviews to ensure the platform's stability and performance. Our team addressed any feedback or issues identified during testing and made necessary integrations.



DEPLOYMENT PHASE

01 Preparation

The necessary infrastructure and environments were set up, including development, staging and production.

02 Deployment Planning

We have created a pitch-perfect deployment plan outlining the steps and procedures for deploying the platform to the production environment.

03 Release Management

Our team deployed the platform to the product environment following the deployment plan. It involved deploying code, configuring servers, and ensuring all dependencies were met.

04 Monitoring and Optimisation

After deployment, our team continuously monitored the platform for performance, security & stability. We promptly addressed any issues or anomalies and made necessary changes.

05 Post-Deployment Review

We conducted a post-deployment review to assess the deployment process's success and gather user feedback. Additionally, our team documented any lessons learned for future deployments.

PROJECT METHODOLOGY

Our team adhered to an Agile methodology during this project, fostering efficient and iterative development. We structured our workflow around sprints, each lasting two weeks, allowing us to focus on specific features and functionalities. Regular feedback sessions with the client, occurring after every sprint, were integral to our process. It ensured our work aligned with the client's evolving requirements and expectations.

Additionally, we employed the project management tool Trello to streamline collaboration and task management, facilitating transparent communication and real-time progress tracking. These practices enabled us to maintain a dynamic and responsive development approach, ultimately delivering a high-quality solution that effectively met the client's needs.

TIMELINE

01 **Total months:** 9 months

02 **No. of Resources:** 7 Resources

03 **Experience of Resources:** 2 Frontend - 4 years

2 Backend - 4 years

2 QA - 4 years

1 Project Manager - 7 years