Door Face Panels

Smart Door Anomaly Detection & Cybersecurity for Elderly Care

CSIS 4495-003 – Applied Research Project

Team Members:

Advitiya Sharda	[300395470]
Eric Sanjo	[300395898]
Reubin Chatta	[300394193]

Team Lead: Advitiya Sharda

Section: 003
Date: January 26, 2026

GitHub Repo: https://github.com/advitiyasharda/W26_4495_S3_AdvitiyaS

Riipen Partner: Door Face Panels (Coquitlam, BC)
Contact: Armin Ghauforian, Founder and CEO
Project Focus: IoT-based smart door security system combining anomaly detection and cybersecurity for elderly care and safety monitoring

1. Introduction

1.1 Domain Overview and Context

With an aging population, there's a growing need for technologies that ensure elderly residents can live independently while remaining safe. Door Face Panels, based in Coquitlam, BC, focuses on modular smart door systems for this purpose, incorporating IoT for access control, anomaly detection, and cybersecurity.

1.2 Problem Framing and Research Questions

Key Challenges:

1. **Fall and Inactivity Detection**: Prolonged absence (24+ hours) may indicate a fall or health emergency
2. **Caregiver Accountability & Compliance**: Transparent logs of who accessed the home and when; compliance with privacy regulations (PIPEDA, GDPR)
3. **Anomalous Behavior Detection**: Identifying unusual access patterns (wandering in dementia, excessive entries/exits, unusual times)
4. **Cybersecurity Threats**: Detecting and responding to unauthorized access attempts, threat actors, unauthorized API usage
5. **Privacy-Preserving Architecture**: Securing facial recognition data and access logs on edge devices without exposing PII

Research Questions:

- **RQ1**: How can we design a system that combines facial recognition access control with machine learning anomaly detection for elderly safety?
- **RQ2**: What threat detection rules and ML models can identify both behavioral anomalies (health emergencies) and security anomalies (unauthorized access) in real-time?
- **RQ3**: How can we implement privacy-preserving, secure edge-based processing while maintaining compliance with data protection regulations?
- **RQ4**: What deployment strategies enable efficient anomaly detection models on resource-constrained hardware (Raspberry Pi/Jetson)?

1.3 Literature Review and Knowledge Gaps

Prior research shows effectiveness of facial recognition using OpenCV and TensorFlow Lite. Isolation Forest and LSTM autoencoders are widely used for time-series anomaly detection. IoT security literature supports lightweight rule-based detection on edge devices. However, limited research integrates behavioral anomaly detection and cybersecurity threat detection into a single edge-based elderly-care system.

1.4 Project Assumptions and Benefits

Assumptions:

- Door usage patterns correlate with resident well-being
- Unsupervised ML can identify abnormal behaviour
- Edge-based processing improves privacy

Benefits:

- Early detection of emergencies
- Improved caregiver awareness
- Privacy-preserving monitoring
- Product differentiation for partner

2. Proposed Research Methodology

2.1 Research Design and Objectives

This project employs an **integrated systems engineering approach** combining facial recognition, machine learning anomaly detection, threat detection, and secure logging with compliance considerations.

Core Objectives:

1. **Facial Recognition & Access Control**: Identify residents and caregivers with <500ms inference latency; log all entry/exit events
2. **Anomaly Detection (ML-Based)**: Train models (Isolation Forest, LSTM autoencoder) to detect unusual door usage patterns (health emergencies, wandering, inactivity)
3. **Threat Detection (Rule-Based & ML)**: Flag suspicious patterns (repeated failed attempts, unauthorized access, unusual times, anomalous usage frequency)
4. **Caregiver Dashboard**: Web-based interface displaying logs, alerts, analytics, and compliance metrics
5. **Security & Compliance**: Encrypted local storage, audit logging, PIPEDA compliance, access controls
6. **Edge Deployment Strategy**: Document how anomaly detection models run efficiently on Raspberry Pi/Jetson without cloud dependency

2.2 Methodology and Justification

- **ML Models**: Isolation Forest for unsupervised anomaly detection.
- **Facial Recognition**: OpenCV, TensorFlow Lite for edge device inference.
- **Threat Detection**: Rule-based approach for suspicious access patterns.
- **Edge Deployment**: All processing done locally (Raspberry Pi).

2.3 Data and Methods

Data Sources:

1. **Facial Recognition Training**: Pre-trained models (OpenCV, TensorFlow Lite) fine-tuned on 3-5 residents/caregivers.
2. **Synthetic Access Pattern Dataset**: Simulated daily door events (~2,000–3,000 events). Data will include normal access patterns (e.g., check-ins, outings) and anomalous patterns (e.g., inactivity, wandering, late-night access).
3. **Real-time Testing**: Test data will be collected from team members and domain experts to validate model performance in real-time conditions.
4. **Techniques**:
5. **Facial Recognition**: Using OpenCV for face detection and feature comparison. No training is required, as transfer learning with pre-trained models will be employed.
6. **Anomaly Detection**:
   - **Isolation Forest**: Unsupervised anomaly detection algorithm used for time-series data, ideal for edge deployment.
   - **LSTM Autoencoder (optional)**: If time permits, the LSTM model will capture temporal dependencies in access patterns to detect anomalies.
7. **Threat Detection**:
   - Rule-based detection will flag anomalous access patterns based on frequency, time-of-day, and unusual behavior.
   - Failed recognition attempts (>3 in 10 minutes) will trigger a threat alert.
8. **Performance Metrics**: Precision, recall, F1-score, and latency will be used to evaluate model performance and optimize for edge device deployment (e.g., Raspberry Pi).

2.4 Technologies and Technology Stack

- **Hardware:** Raspberry Pi 4, Nvidia Jetson Nano (optional), CSI Camera for facial recognition.
- **Programming:** Python, OpenCV, TensorFlow Lite, scikit-learn, Flask for API, SQLite for local logging.
- **Front-End & Back-End:** Flask API, simple HTML/CSS/JS dashboard, SQLite for data logging.

2.5 Expected Results and Practical Applications

Expected Outcomes:

1. **Facial Recognition Module**:
   - 85% accuracy on custom dataset (3-5 residents)
   - <500ms per-frame latency on Raspberry Pi
   - Real-time processing from camera feed

2. **Synthetic Dataset**:

- 2,000-3,000 labeled access events representing 1-3 months
- Clear separation of normal vs. anomalous patterns
- Validated by domain expert (Armin) for realism

3. **Anomaly Detection Models**:

    - **Isolation Forest**: >80% F1-score on synthetic test set
    - **LSTM Autoencoder** (optional): >85% F1-score if time permits
    - <200ms inference latency on Raspberry Pi
    - Validated on real-time test data (team member access patterns)

4. **Threat Detection**:

    - Flags repeated failed recognition attempts (>3 in 10 min)
    - Detects prolonged absence (>24 hours without exit)
    - Alerts on unusual access times and frequency spikes
    - Distinguishes health alerts vs. security alerts

5. **Caregiver Dashboard**:

    - Displays current and historical access logs with anomaly scores
    - Shows active health alerts (inactivity) and security alerts (threats)
    - Provides trend visualization (when anomalies occur, frequency patterns)
    - Compliance report: audit trail of all system actions, data handling summary

6. **Documentation and Strategy**:

    - Well-organized GitHub repository with clear folder structure
    - README with setup, training, and deployment instructions
    - Model documentation: architecture, hyperparameters, performance metrics
    - Deployment strategy: how to run models efficiently on edge devices
    - Security & compliance guide: PIPEDA considerations, data handling, audit logging
    - API documentation with examples

Practical Applications:

- **Prototype Deployment**: Controlled test environment (lab or partner site)
- **User Feedback**: Gather input from Door Face Panels and potential end-users (care facilities)
- **Compliance Validation**: Demonstrate PIPEDA-compliant approaches
- **Future Phases**: Add voice analysis, cloud integration, mobile alerts, expanded model types

3. Riipen External Partner

Partner Organization: Door Face Panels
Location: Coquitlam, British Columbia, Canada
Website: https://thedoorface.com
Company Size: 2–10 employees (startup)

Primary Contact:

- **Name**: Armin Ghauforian
- **Title**: Founder and CEO
- **Email**: doorface.panels@gmail.com

Deliverables for Door Face Panels:

- Facial recognition + access logging module (working code)
- Trained anomaly detection models (Isolation Forest, optionally LSTM)
- Synthetic dataset and threat detection rules
- Web dashboard for caregiver monitoring
- GitHub repository with complete documentation
- Deployment strategy and performance metrics
- Security & compliance documentation
- Demo/presentation of working system

4. Project Planning and Timeline

4.1 Project Timeline (Revised Timeline - Topic Selection Mid-January)

Week 0: Project Ideation & Selection (Jan 6 – Jan 19)

- Team brainstormed and evaluated multiple Riipen projects
- Discussed Door Face Panels projects (anomaly detection + cybersecurity)
- Decided to combine both projects for richer scope
- Initial meeting with Armin Ghauforian to align expectations and priorities
- **Hours per person**: ~4-5 hours (planning, meetings, research)

Phase 1: Research & Setup (Weeks 1–3, Jan 20 – Feb 2)

- Detailed project planning and role assignments
- Research facial recognition libraries, anomaly detection algorithms, threat modeling
- Set up development environments (Python, Flask, OpenCV, scikit-learn, SQLite)
- Initialize GitHub repository with proper folder structure
- Design database schema and API endpoints
- **Deliverables**: Environment ready, GitHub repo created, project plan finalized
- **Hours per person**: ~10-12 hours

Phase 2: Core Implementation (Weeks 4–9, Feb 3 – Mar 16)

Advitiya's Focus (Cybersecurity & Facial Recognition):

- Implement facial recognition pipeline (camera → OpenCV/TensorFlow detection → logging)
- Build Flask API for facial recognition inference and threat detection
- Design SQLite schema for access logging and audit trails
- Implement threat detection rules (repeated failures, unusual times, frequency spikes)
- Security hardening (encryption, access control, audit logging)
- **Checkpoint (Feb 24)**: Working facial recognition + basic access logging
- **Est. hours**: 18-22h

Eric & Reubin's Focus (Data & Anomaly Detection):

- **Eric**: Generate synthetic door access dataset (1-3 months, 2,000-3,000 events)
  - Design normal vs. anomalous patterns
  - Implement data generation scripts
  - Validate dataset realism with Armin

- **Reubin**: Implement anomaly detection models

  - Train Isolation Forest on synthetic data
  - Evaluate model performance (precision, recall, F1)
  - Optimize for edge deployment (memory, latency)
  - Support facial recognition integration

- **Collaborative**: Develop threat detection rules combining security + behavioral anomalies

- **Checkpoint (Feb 24)**: Working dataset + baseline anomaly detection model

- **Est. hours per person**: 16-20h

Deliverables by End of Phase 2: Progress Reports 1 & 2; working facial recognition + access logging + basic anomaly detection

Phase 3: Integration & Testing (Weeks 10–12, Mar 17 – Apr 2)

All Team Members:

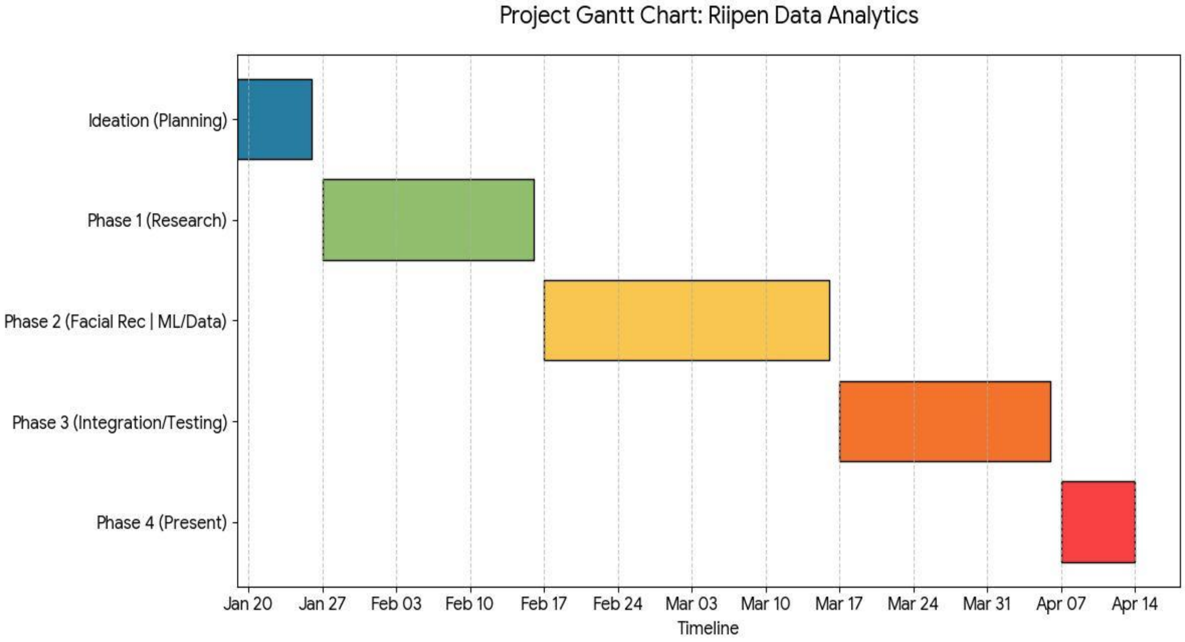
- Integrate facial recognition, anomaly detection, and threat detection into unified system
- Build caregiver dashboard interface
- Develop analytics and compliance reporting
- End-to-end testing on real hardware (Raspberry Pi)
- Performance optimization (latency, memory, accuracy)
- Security hardening and vulnerability testing
- Documentation finalization (README, API specs, deployment guide, security guide)
- **Milestone**: Final Checkin (Mar 24) – complete system demo
- **Deliverables**: Progress Reports 3, 4, 5; final codebase; documentation
- **Est. hours per person**: 12-16h

Phase 4: Presentation (Week 13, Apr 7 – Apr 14)

- Final Report preparation and polishing
- Presentation slides and demo planning
- Live demonstration on hardware
- **Deliverable**: Final Report + Presentation
- **Est. hours per person**: ~5h

Total per person: ~55-65 hours (with Phase 0 planning: ~60-70 hours, realistic target)

4.2 Revised Timeline (Gantt Chart - From Jan 19 Selection)



4.3 Collaborative Team Responsibilities

Advitiya Sharda (Team Lead) – Cybersecurity, Facial Recognition & System Architecture

Primary Responsibilities:

- Lead cybersecurity threat modeling and threat detection design
- Implement facial recognition pipeline (camera → OpenCV detection → logging)
- Build Flask API for facial recognition and threat detection endpoints
- Design SQLite schema for secure logging and audit trails
- Implement threat detection rules (security-focused: repeated failures, unauthorized access)
- Security hardening (encryption at rest, access control, audit logging)
- Coordinate system architecture and integration with Eric/Reubin's ML work
- Lead all joint submissions and Riipen partner communication

Eric Sanjo – Data Science & Anomaly Detection (Dataset)

Primary Responsibilities:

- Design and generate synthetic door access dataset (realistic elderly care patterns)
- Create normal vs. anomalous patterns (health emergencies, security threats)
- Validate dataset with Armin for realism
- Support threat detection rule development with data insights
- Contribute to model evaluation and performance testing

- Document data generation methodology and rationale

Reubin Chatta – Anomaly Detection & System Integration

Primary Responsibilities:

- Train and evaluate anomaly detection models (Isolation Forest, optionally LSTM)
- Implement data preprocessing and feature engineering
- Optimize models for edge deployment (latency, memory)
- Build web-based dashboard (HTML/CSS/JavaScript)
- Integrate all modules (facial recognition + anomaly detection + threat detection)
- System testing and end-to-end validation
- Documentation (README, API specs, deployment guide)

4.5 Work Time Breakdown

Phase	**Advitiya**	**Eric**	**Reubin**	**Team Total**
Week 0 (Ideation/Planning)	1.5h	1.5h	1.5h	4.5h
Phase 1 (Research/Setup)	4h	3.5h	3.5h	11h
Phase 2 (Implementation)	14h	10h	12h	36h
Phase 3 (Integration/Testing)	6h	4h	8h	18h
Phase 4 (Presentation)	2h	1.5h	1.5h	5h
Total	**27.5h**	**20.5h**	**26.5h**	**74.5h**

5. Project Contract

We, the undersigned members of this project team, agree to the following terms to ensure a successful and collaborative project experience:

1. **Meetings**: We will hold at least one mandatory in person meeting per week (normally being just after class on Tuesday) to discuss progress, resolve issues, and plan next steps. Additional meetings will be scheduled as needed.
2. **Communication**: Our primary channel for communication will be Discord. For official documentation and urgent matters, we will use our student emails. We commit to responding to team communications within 24 hours.
3. **Task Management**: We will use a shared GitHub repository for code, documentation, and tasks tracking. Each member is responsible for keeping their assigned tasks updated.
4. **Commitment**: We both commit to logging our work hours accurately , as per the course guidelines.

Name: Advitiya Sharda Date: 26th January, 2026

Name: Eric Sanjo Date: 26th January, 2026

Name: Reubin Chatta Date: 26th January, 2026

6. AI Use Section

6.1 Table of AI Tools and Specific Use

AI Tool Name	**Version & Account Type**	**Specific Feature / Purpose**	**Value Addition**
ChatGPT	GPT-4, Free Trial	Initial Research about various features and technologies used in the project . Project report creation help	Reviewed outputs for accuracy, Learned and used it in the report creation

6.3 AI Prompt History

Representative prompts used during project preparation:

Technical Implementation:

- "What are the technologies used in edge computing for facial recognition systems"

Documentation:

- "How to structure a technical project proposal for initial project proposal based on the proposal below: ”
- "How to setup a well working github repo based on the project requirements"

7. Work Date/Hours Logs

Date	Student Name	Hours	Description of Work
Jan 19, 2026	Advitiya Sharda	1	Initial team meeting :reviewed Riipen brief, scanned available projects, and discussed possibly proposing our own cybersecurity-focused idea.
Jan 19, 2026	Eric Sanjo	1	Initial team meeting: compared multiple Riipen options
Jan 19, 2026	Reubin Chatta	1	Initial team meeting: helped shortlist 2–3 candidate project directions
Jan 20, 2026	Advitiya Sharda	1	Follow-up team discussion: narrowed ideas to; checke feasibility against course timeline.
Jan 20, 2026	Eric Sanjo	1	Follow-up team discussion: considered data availability
Jan 20, 2026	Reubin Chatta	1	Follow-up team discussion
Jan 21, 2026	Advitiya Sharda	1.5	Joint meeting with Armin (Door Face Panels): discussed initial ideas for smart door security and anomaly detection. Note taking
Jan 21, 2026	Eric Sanjo	1.5	Joint meeting with Armin: gathered requirements, clarified expectations.

Date	Student Name	Hours	Description of Work
Jan 21, 2026	Reubin Chatta	1.5	Joint meeting with Armin: took notes on partner priorities, dashboard expectations, and hardware assumptions.
Jan 21, 2026	Advitiya Sharda	2	Internal team discussion: compared alternative project ideas (forensics, dashboards, smart home), agreed on focusing on smart door security + cybersecurity/compliance.
Jan 21, 2026	Eric Sanjo	2	Internal team discussion: evaluated feasibility of anomaly detection vs. purely rule-based approach; agreed to keep ML as comparative study.
Jan 21, 2026	Reubin Chatta	2	Internal team discussion: considered front-end complexity, deployment constraints, and how to keep dashboard scope realistic.
Jan 25,2026	Advitiya Sharda	2	Team meeting: Project proposal crafting meeting
Jan 25,2026	Reubin Chatta	2	Team meeting: Project proposal crafting meeting
Jan 25,2026	Eric Sanjo	2	Team meeting: Project proposal crafting meeting
Jan 26, 2026	Advitiya Sharda	5	Project proposal crafting (including 2 hour meeting)

Date	Student Name	Hours	Description of Work
Jan 26, 2026	Reubin Chatta	3	Project proposal crafting + Team meeting
Jan 26, 2026	Eric sanjo	2.5	Project proposal crafting + Team meeting

8. Closing and References

8.1 Acknowledgments

We acknowledge the invaluable support of:

- **Armin Ghauforian** (Founder & CEO, Door Face Panels) for domain expertise, project guidance, feedback on priorities, and validation of synthetic data realism
- **Priya** (Instructor, CSIS 4495, Douglas College) for guidance on research methodology, proposal requirements, and course expectations
- **Douglas College** for access to computing resources, library services, and learning support
- **Riipen Platform** for facilitating collaboration with industry partners and real-world project opportunities

8.2 References

[1] Sixsmith, A., & Sixsmith, J. (2008). "Ageing in Place in the United Kingdom." *Ageing International*, 32(3), 219–235.

[2] Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media.

[3] Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). "Isolation Forest." *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM '08)*, 413–422.

[4] Thirumurugan, P., et al. (2019). "Anomaly Detection in Smart Homes Using Isolation Forest." *IEEE IoT Journal*, 6(4), 7410–7419.