Low-cost Smart Sensor Spoofer on an Unmanned Air System

S26-04

Izzy Burley, Will Field, Carl Holmberg, Kush Parmar, Drew Schineller, and Sachel Jetly

12/4/2025

COLLEGE OF ENGINEERING
BRADLEY DEPARTMENT OF
ELECTRICAL AND COMPUTER ENGINEERING
VIRGINIA TECH

Product Description

Project Overview

- Develop a dual-UAS system that **generates**, **spoofs**, and **detects** multi-sensor data (LiDAR, camera, GPS).
- Provide the Navy with a **low-cost, UAV spoofing platform** for **research and training**.
- Demonstrate **real-time spoofing detection and classification** using **onboard neural networks**.

Product Components

- Have a **ground-based spoofing module** to simulate LiDAR, Radar, GPS, or camera spoofing with a **point cloud** in a simulated, Unreal Engine (UE5), **virtual environment**.
- Integrate a **neural network processing unit** on the UAV for real-time **spoof detection** and **classification**.
- Designing a **control and navigation subsystem** in **Simulink** to maintain stable flight and compensate for spoofed data inputs and provide a connection for **data transfer**.
- Creating a **ground control station (GCS) interface** for operator **monitoring, data logging, and configuration of spoofing profiles**.

Problem Statement

- Our customer requires **data collection and analysis** on sensor spoofing and spoof detection between **two UAS platforms** equipped with sensor and controlled emitter hardware.

- This project will aid in the research and development of **small, inexpensive UAV spoofer platforms** to be used by the US Navy and increase awareness of **spoofing techniques** and **neural network applications**.

- The final project will be completed Spring 2026 and will produce desired datasets (real and spoofed) and **trained detection models** through a simulated & physical UAV system.

Problem Impact

- Our project has significant potential impacts, as LiDAR and Radar spoofing are **dangerous cyberattacks** that can **damage** autonomous vehicles, devices, and users.
- All testing and operations of emitters will be performed in an anechoic chamber and abide by FCC and FAA regulations, **avoiding potential risks**.
- The **data** produced by our project is **deemed valuable** by our customer and will benefit the warfighter greatly.

Primary Stakeholders

- *NAWCAD Pax River* (Primary Stakeholder/Customer)
- *US DoD* (Research data for inexpensive spoofer UAS)
- Foreign Military Forces (Work may impact foreign allies/enemies)
- *VT AOE* Department (Possible UAV source)

COLLEGE OF ENGINEERING
BRADLEY DEPARTMENT OF
ELECTRICAL AND COMPUTER ENGINEERING
VIRGINIA TECH

Technical Requirements & Constraints

High Level Requirements

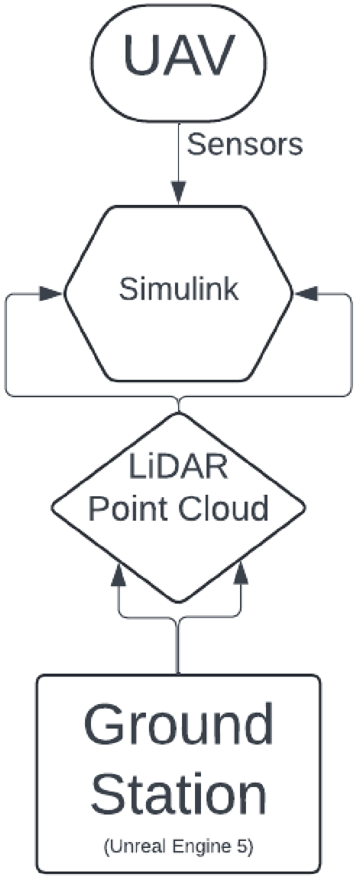
- Capability to **spoof a targeted sensor**
- Mimic **environmental** conditions
- **Log** and **process** sensor input
- **Air gapped edge neural network** with hardware inputs

High Level Constraints

- **Existing** LiDAR spoofing **technologies**
- **Spoofing risks** involving civilian autonomous systems
- **Budgeting** and **time** limitations on a complex project

System Design - High Level

- 3 Distinct subsystems: Ground Station, Simulink Bridge, UAV & Neural Network
- Ground Station, virtualized in **Unreal Engine 5,** generates LiDAR point cloud
- Virtual Drone linked to physical UAV via Simulink "bridge"
- UAV sensors injected with data from Ground Station.
    - Neural Network classifies accordingly



COLLEGE OF ENGINEERING
BRADLEY DEPARTMENT OF
ELECTRICAL AND COMPUTER ENGINEERING
VIRGINIA TECH

System Interaction

Data Flow

Ground station → Simulink Bridge → UAV sensor array → Neural Network analysis → classified results

via Ground Station logging

Power

- UAV is powered by its **onboard battery system** and supports all flight and computation subsystems, including the neural network hardware.
- Ground Station and Simulink Bridge are powered by **ground-based systems.**

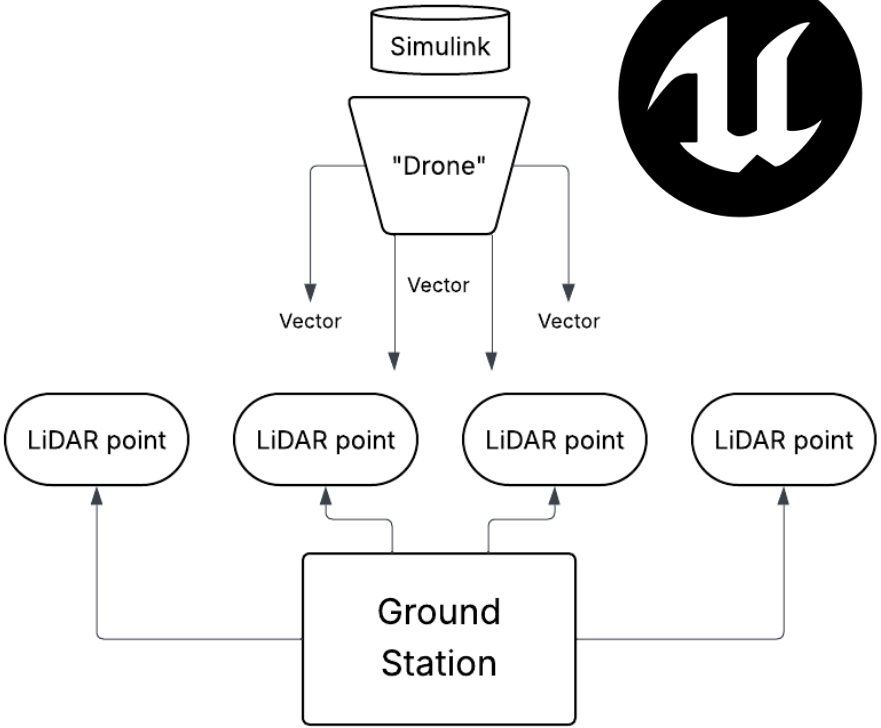
Controls

- Commands for **flight control and spoofing configuration** come from the Ground Station.
- The **UAV's control system** autonomously adjusts its response based on the **neural network's classification outputs** (e.g., if spoofing is detected, the UAV can trigger a failsafe mode or recalibration).

COLLEGE OF ENGINEERING
BRADLEY DEPARTMENT OF
ELECTRICAL AND COMPUTER ENGINEERING
VIRGINIA TECH

Architectural Design - Virtual Ground Station

- All takes place in **Unreal Engine 5**
  - What is Unreal Engine?
- Mitigates project **cost**, **accelerates** timelines, & allows control
- Easily manipulated environment with deployable parameters



Physical Design - Virtual Ground Station

Physical Constraints

- **Stationary** system (portability not required)
- Runs **Unreal Engine** + **Simulink** on a single computer
- Minimal mechanical concerns (no shock/vibration requirements)

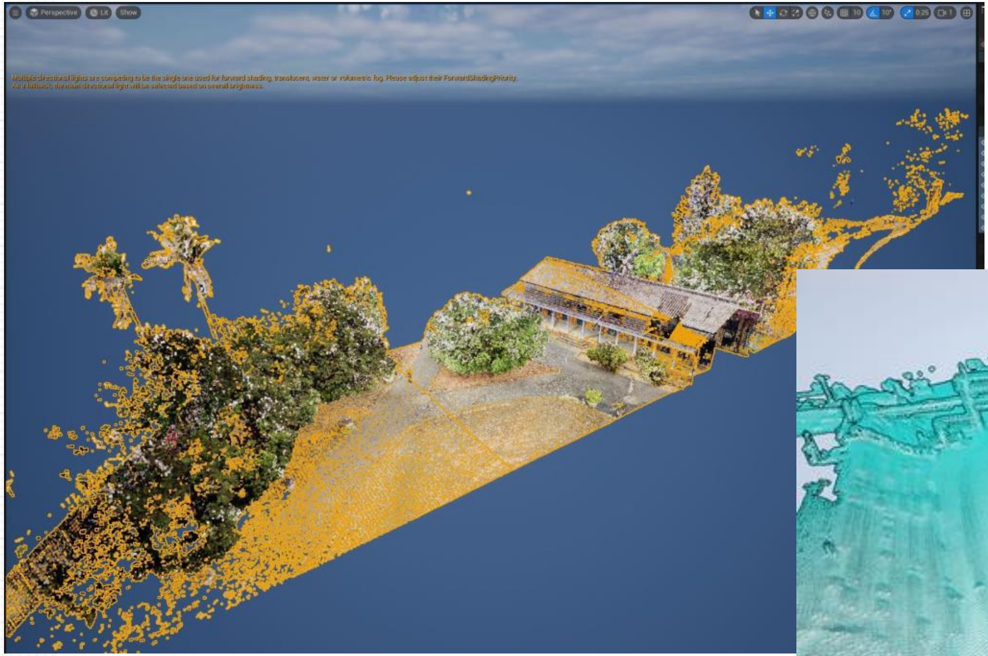
Environmental Considerations

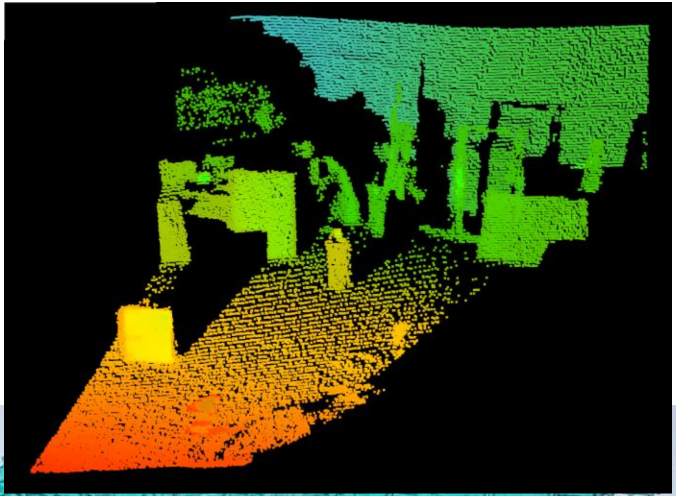
- Main risks: temperature + humidity affecting **computer operation**
- If used **outdoors**, requires a **weather-resistant enclosure** (Pelican case)
- **Indoor** HiL testing only needs a **standard computer** capable of UE5 + Simulink

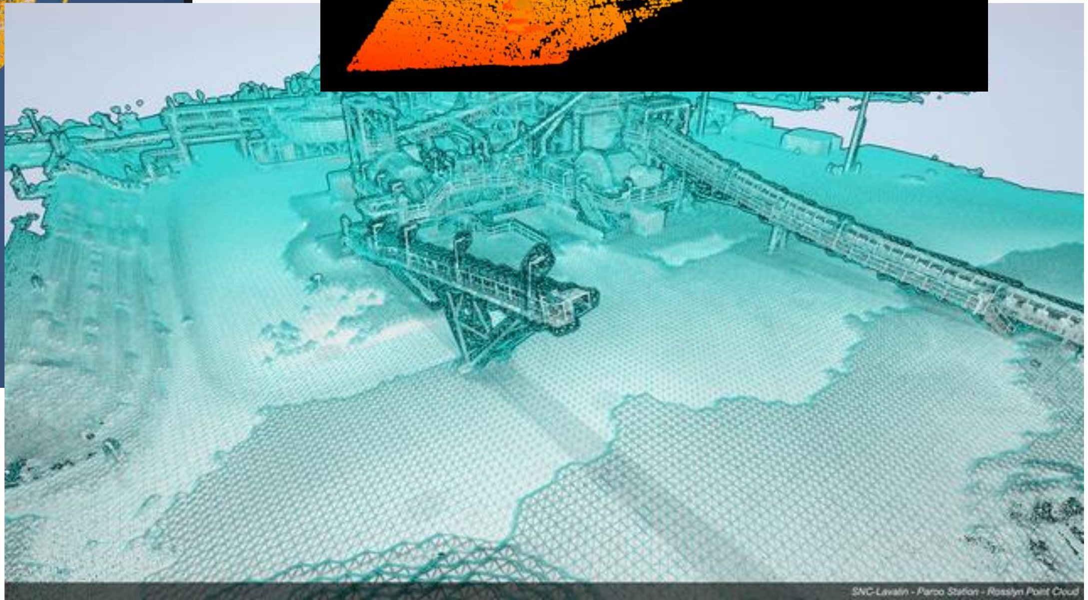
Design Implications

- Ensure proper **cooling and airflow**
- **Protect sensor hardware** from moisture when deployed in field testing

Unreal Engine 5







LiDAR Point Cloud - Virtual Ground Station (UE5)

Raycast per beam → collect hit points → inject spoof points → timestamp → sort → output frame

Virtual LiDAR Environment

- Entire LiDAR data is **generated** inside Unreal Engine 5
- UE5 **simulates “real”** and **“spoofed”** LiDAR using **ray-based scene sampling**

How the Scan Is Produced

- Virtual environment built with **3D models, materials, and lighting**.
- A virtual **LiDAR model** defines **beams, FOV, rotation rate, and resolution**.
- **UE5 raycasts** from the sensor pose to **compute depth, intensity, and geometry hits**.
- **Spoofing elements** (false walls, ghost points, injected geometry) **generated** using the **same ray engine.**

Output & Integration

- Each **frame** exported as a **time-tagged point cloud** with **per-point metadata**.
- **Data** is **serialized** and sent to **Simulink for neural-network processing**.

COLLEGE OF ENGINEERING
BRADLEY DEPARTMENT OF
ELECTRICAL AND COMPUTER ENGINEERING
VIRGINIA TECH

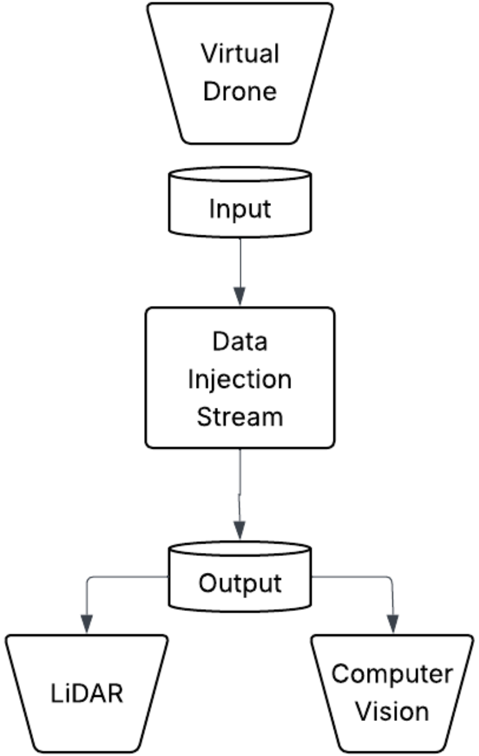
LiDAR - FMCW & Pulsed ToF

FMCW: linear frequency modulated sweeps using a local oscillator. This type of LiDAR is more resilient and can have various chirp patterns. It is demodulated by using the beat frequency of the chirps and solves for range and radial velocity. This is an end goal for a resilient spoofing system.

Pulsed ToF: transmission of short pulses and measuring return time and intensity/scattering. This is what is modeled in UE5 and our sourced LiDAR sensor uses.

Architectural Design - Simulink Bridge

- Acts as **middleware** between Unreal Engine & UAV/Neural Network
- Allows for **direct injection** of sensor data from virtual to physical
- Introduces seamless transfer of **unformatted to formatted data**



Algorithms & Flow Control – Simulink Bridge

Communication Layers

- **Input:** TCP client socket receiving LiDAR frames from UE5.
- **Internal:** Simulink bus structures carrying LiDAR Frame (timestamp, pose, point array, metadata).
- **Output:** UDP or Serial (RS-232/TTL) link to edge hardware.

Error Handling

- CRC verification
- Packet counters
- Acknowledgment scheme to detect drops/duplicates

Tools & API Integration

- MATLAB API for debugging and visualization: *sendFrame(), readResult(), checkCRC()*
- Allows rapid testing without recompiling Simulink models.

COLLEGE OF ENGINEERING
BRADLEY DEPARTMENT OF
ELECTRICAL AND COMPUTER ENGINEERING
VIRGINIA TECH

Performance Requirements - Simulink Bridge

- Sustains 10–30 Hz LiDAR throughput.
- Maintains deterministic timing for downstream analysis.
- Matches real sensor baud rates and capabilities

<u>Message Data Structure</u>

LiDARHeader:

- frame_id
- timestamp (ns)
- point count
- flags
- CRC-32 integrity check

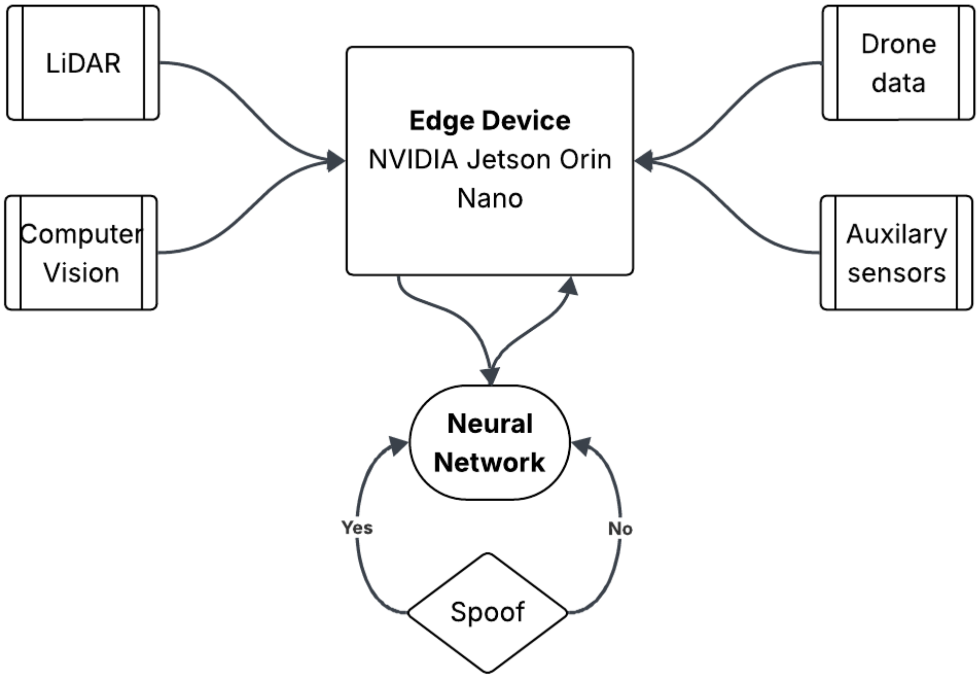
Payload:

- Packed LiDAR points (x, y, z, intensity)

COLLEGE OF ENGINEERING
BRADLEY DEPARTMENT OF
ELECTRICAL AND COMPUTER ENGINEERING
VIRGINIA TECH

Architectural Design - UAV & Neural Network

- Deployable **air gapped** neural network
- Devices interface with **drone bus**
- Hardware + software "networks"
- **Drone requirements**:
  - Signal sources
  - Edge device for neural network
  - Payload capacity



Physical Design – UAV & Neural Network

Environmental Constraints

- **Fully exposed** to vibration, temperature, and humidity during flight
- Must protect **Jetson Orin Nano, Raspberry Pi Pico, battery,** and **sensors**
- Electronics are bare PCBs → require **full enclosure** and insulation

Design Considerations

- **Weight** is critical and determines which UAVs can **carry the payload**
- Final weight and **power requirements** depend on the chosen LiDAR/GPS modules
- Only **essential interfaces** and **sensors** (LiDAR, GPS if external) should remain exposed

System Implications

- Enclosure ensures **durability** and environmental protection
- Allows the NN module to function as a standalone, **UAV-agnostic payload**
- **Improves portability** and **simplifies integration** with different UAV platforms

Inference & Decision Logic - Neural Network

Real-Time Inference Pipeline

- **Jetson** receives each **LiDAR frame** from **Simulink** and performs on-device **spoof detection**.
- **Frames** are **converted** into model-ready features (voxel grid or point features).
- Quantized **neural network** (TensorRT/ONNX) runs inference and **outputs a spoof likelihood**.

Preprocessing

- **Voxelization** or **downsampling** to reduce compute load.
- **Normalization** to match training distribution.
- Optional feature augmentation: ring index, range, curvature.

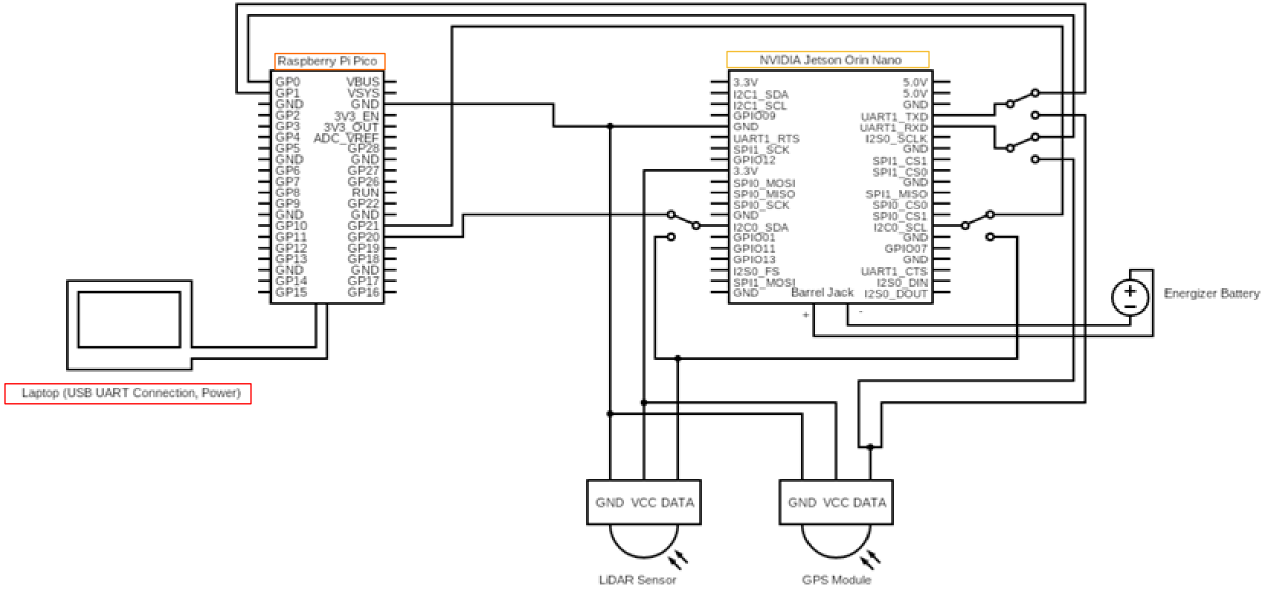
Decision Logic

- **Spoof score** smoothed over time (**exponential filter**).
- **Hysteresis thresholds** used to prevent rapid switching.
- Output includes: **flag (real/spoof), score, model metadata, latency, health status**.

Fault Handling

- If timing or inference errors occur, system falls back to a lightweight baseline model or rule-based detector.
- Marks output as **degraded** while maintaining real-time operation.

Physical Wiring Diagram - System



- **Laptop** sends **spoofed LiDAR** (I²C) and **GPS** (UART) signals **via Raspberry Pi Pico**
- Signals injected into **Jetson Orin Nano** for **edge-AI processing** and **logging**
- Hardware switches **select real vs. spoofed** sensor inputs
- Jetson **powered** by a **10,000mAh** portable battery

Timings & Loop Rates - System

Stage	Rate (Hz)	Deadline (ms)	p95 Latency	Notes
UE 5 LiDAR Gen	20-30	33-50	<= 80%	Raycast + header pack
Simulink	20-30	10	<= 6%	CRC, schema, ID, forwarding
Edge Preprocess	20-30	8	<= 5%	Downsample & normalize
Edge NN Inference	20-30	10	<= 8%	Quantized engine
Postproc + Publish	20-30	5	<= 3%	Smoothen metrics

- Operates in fixed cycles at a target frame rate (Hz)
- A cycle = one **LiDAR frame** from UE5, **transport via Simulink**, **preprocessing, neural inference** on the **edge device**, and **feedback**
- Components run asynchronously

COLLEGE OF ENGINEERING
BRADLEY DEPARTMENT OF
ELECTRICAL AND COMPUTER ENGINEERING
VIRGINIA TECH

Live Demo

Subsystem	Need	Engineering Characteristic	Target Specifications	Test Case(s)
Spoofing Module	Generate virtual real and spoofed LiDAR point clouds and GPS frames	-Point cloud generation -Spoof insertion - Stable frame rate	• Mimics buildings/terrain objects (REQ-SP-001) • Supports ≥2 sensor types (LiDAR + GPS) (REQ-SP-002) • Adjustable spoof profiles (REQ-SP-003/004) • 10-min continuous spoof profile (PERF-SP-002) • Spoof adapts <30s (PERF-SP-003)	**Test 1:** Baseline – Real signal transmission **Test 2:** Variant – Spoofed signal transmission **Test 3:** Variant – Combined real + spoofed transmission
NN Module	Classify spoofed and real sensor data in real time	Real time signal classification accuracy; model adaptability	• Real-time NN analysis (REQ-NN-001) • ≥85% accuracy (PERF-NN-002) • Multi-model support (REQ-NN-003) • Retrain <30s (PERF-NN-003) • Adversarial detection (REQ-NN-006)	**Test 1:** Baseline – Recognition of real signals **Test 2:** Variant – Recognition of spoofed signals **Test 3:** Variant – Mixed (spoofed + real) signal detection
Control System Module	Manage the timing, formatting, and delivery of LIDAR/GPS frames through the whole system (UE5 to Jetson)	Data formatting, timing control, adaptive recalibration; failsafe response	• Selectively spoof certain sensor streams (REQ-SP-004) • Supports wireless profile updates (REQ-SP-006) • Logs real/spoof/control data (REQ-CN-004) • Stable data transfer for NN and logging (system requirement)	**Test 1:** Baseline – Flight stability **Test 2:** Recalibration under gradual spoofing
Ground Station Module	Configure the spoofing profiles and monitor the communication and recording	Data logging, data integrity; spoof selectivity; point cloud selectiveness	• Spoof profile/config UI (REQ-SP-005) • Logs real + spoofed + NN data (REQ-SP-007) • Displays sensor data to operator (Must-Have) • Reliable link ≥95% packet delivery (original GCS spec)	**Test 1:** Baseline – Communication link/data integrity **Test 2:** Selective spoofing control across sensors

Target Specification & Testing

Modules/Subsystems

- Work breakdown structure (WBS)
- Engineering characteristic
- Verification and Validation (prev. Slide 14)

Responsible Engineer	
Will/Carl	(1)
Kush/Sachel	(2)
Kush/Drew	(3)
Izzy/Carl	(4)
Will/Drew	(5)
Izzy/Sachel	(6)

	Module	Requirement ID(s)	Verification Method
(1)	Spoofing Module	FUN-1, FUN-3, FUN-4, FUN-5, FUN-7, FUN-8, FUN-9, CON-2	Flight test, vary parameters, isolation testing, exposure to different scenerios
(2)	Neural Network Module	PER-1, PER-2, PER-3, PER-4, PER-5, I/O-1, CON-3	Continuous collection, latency testing
(3)	Control System	COM-1, COM-2, ENV-1, I-O3, TEST-1, CON-1	Documentation comparison, Relocation, Hardware, Chamber for flight
(4)	Ground Control Station	I/O-2, TEST-2	Identify reliable product, Measure distance and test
(5)	Power Module	POW-1, ME-1	Measure in stable state
(6)	Failsafe Module	REL-1, REL-2	Observe flight

COLLEGE OF ENGINEERING
BRADLEY DEPARTMENT OF
ELECTRICAL AND COMPUTER ENGINEERING
VIRGINIA TECH

Schedule

Phase	Dates	Tests	Subsystems	Resources
1a	10/27 - 11/21	Virtual Environment Setup Initial Communication	Spoofing Ground Station	Unreal Engine 5, MATLAB Unreal Engine 5, Simulink, Jetson
1b	11/24 - 12/12	Jetson Setup Begin LiDAR Simulations Neural Network Training	Neural Network, Ground Station, Control System	Jetson, Python, Unreal Engine 5, MATLAB, Simulink, Data Sets, White Papers
2	1/19 - 2/13	LiDAR Point Cloud Vector Array Creation Training System Procurement	Neural Network, Spoofing Full System	MATLAB, Jetson, Python, Unreal Engine 5 Adafruit, Nvidia, VT Drone Groups
3	2/16 - 3/20	Dynamic LiDAR Attack Simulations and Modification	Full System	Unreal Engine 5, Simulink, Jetson, MATLAB, Python, Data Sets
4	3/27 - 4/14	Physical System Testing	Full System	UAV, Jetson, Laser Array
F	4/23	Project Presentation	Full System	UAV, Control System

Budget

Phase Needed	Item	Source	Unit Cost	Quantity	Total Unit Cost
1b*	AI Edge Computer	NVIDIA Jetson Orin Nano Super Developer Kit	$250.00	1	$250.00
2*	LiDAR Sensor Array	VIAVI	$0.00	1	$0.00
3	Microcontroller	Raspberry Pi Pico Microcontroller w/ RP 2040 Chip	$7.00	1	$7.00
3	GPS Module	GPS Module GY-NEO6MV2 (u-blox NEO-6M) + patch antenna Ali Express	$1.66	1	$1.66
4	UAV (Drone)	TBD (AOE Department Loan?)	$0.00	1	$0.00
4	Power Supply	Energizer | MAX 10,000 mAh 15W USB-C 3 Port Bank	$15.59	1	$15.59
				Total	**$274.25**

**INDICATES ORDER HAS BEEN PLACED*

COLLEGE OF ENGINEERING
BRADLEY DEPARTMENT OF
ELECTRICAL AND COMPUTER ENGINEERING
VIRGINIA TECH

Conclusion

Accomplishments

- Sourcing of most COTS componentry for spoofer station and spoof detection UAV
- Research into LiDAR, GPS and Radar spoofing attacks with some documentation
- Identified target dates and milestones for project completion
- Developed scalable high-level project solution that can be adjusted based on budget
- **Built initial virtual sensing environment, LiDAR generation workflows, and communication paths**

Next Steps (*Immediate*)

- Complete **hardware ↔ software logic flow** across UE5 → Simulink → MCU → Jetson.
- Continue **training & refinement** of spoof-detection neural models + research SLMs.
- Finalize remaining **component selections, budget decisions, and integration design**.
- Begin review of **FAA requirements** related to LiDAR/GPS manipulation and controlled testing.
- Document **signal generation standards** for safe, compliant spoofing experiments.

Q & A

Thank you!

Index

Preliminary Resource Planning

Resources Needed	Sources	Critical Items
UAV (≥40 ft, ≥15 mph)	Other Design Teams/ VT Labs	Stable UAV with battery life
Camera (15–25 m, 360° FOV)	Drone sensor vendors	Camera that meets detection spec
Neural network controller (real-time, low-power)	Jetson / FPGA boards	Power-efficient NN processor
MATLAB + ML frameworks	VT license & open-source	2 types of sensors (GPS, LIDAR, Camera (IR or Visible), Rader, Acoustic Sensor) TBD
Laser / rangefinder	COTS laser suppliers	Laser that meets detection spec
Ground station (3-mile range) + Testing Area	VT labs	

Preliminary Information Requirements

- Spoofing technologies? What and how?
- Drone simulation vs practical
- LiDAR vs GPS, how to interact/generate
- Explore FAA & FCC guidelines, explore anechoic chamber facilities
- Understand/experiment neural network deployment, training, and hardware inputs
- Explore edge computing devices, NVIDIA, Qualcomm, Google, Raspberry Pi, etc