Amrita Vishwa Vidyapeetham

Iron Shield Real Time Defense System Using Neural Network

SHREEVARSINII BALASEKAR CB.AI.U4AIM24144

PROBLEM STATEMENT

Problem Background

- Missile defense systems must decide very quickly
- Decision needed:
  I)Intercept the missile OR Ignore it
- A wrong decision can:
  I) Waste defense missiles
  II)Or fail to protect people and important places

Existing Problem

- Current systems mostly use:
  I)Fixed rules
  II) Pre-programmed logic
- These systems:
  I)Cannot handle new or noisy trajectories
  II)Give false alarms
  III)Are slow or heavy for real-time use
- They are not intelligent or adaptive

TECH STACK USED

◆ Programming & Tools

- Python
- NumPy, Pandas
- Matplotlib (trajectory visualization)

◆ Machine Learning

- TensorFlow / Keras
- Feedforward Neural Network (FNN)
- Keras Tuner (hyperparameter optimization)
- Scikit-learn (scaling & evaluation)

◆ Physics Modeling

- Kinematic equations
- Gravity-based trajectory simulation
- Gaussian noise for realism

◆ Deployment

- TensorFlow Lite (TFLite)
- Edge / real-time inference (Streamlit)

DATASET INFORMATION

- A synthetic dataset was generated to simulate missile trajectories.
- Two classes: Intercept (1) and Ignore (0), with a balanced 50:50 split.
- Total dataset size: 20,000 samples.
- Each sample has 6 input features: initial position (x_0, y_0, z_0 in km) and initial velocity (vx, vy, vz in km/s).
- Trajectories modeled with gravity ($g = 0.0098$ km/s²) and Gaussian noise ($\sigma = 0.1–0.5$ km).
- Position equations:
  - $x(t) = x_0 + vx \cdot t + \text{noise}$
  - $y(t) = y_0 + vy \cdot t + \text{noise}$
  - $z(t) = z_0 + vz \cdot t - \frac{1}{2} \cdot g \cdot t^2 + \text{noise}$
- This dataset provides a realistic yet controlled environment for training the model.
- Trajectories within 10 km defense zone → Intercept (1), else Ignore (0).
- Dataset balanced to prevent bias.
- Features scaled before training.
- Data split: 70% train, 20% validation, 10% test.
- Physics-based dataset ensures reliable training.
- Future work: testing with real missile data (DRDO).

PROPOSED WORK

- Goal: Classify missiles → Intercept (threat) or Ignore (non-threat)
- Dataset:
  - 20k synthetic trajectories (6 features: position + velocity)
  - Physics-based, balanced labels (R = 10 km defense zone)
  - Preprocessing: Normalized (StandardScaler), randomized, split (70/20/10)
- Model: Feedforward NN → layers, ReLU + Dropout, Sigmoid output
- Optimization: Keras Tuner (learning rate & layer size), F1-optimized threshold (0.8946)
- Deployment: TensorFlow Lite on edge devices + physics-based override check
- Deployment demonstrated using a Streamlit-based simulated dashboard
- Result: Fast, accurate, resource-efficient real-time defense system

PROPOSED WORK

Insights

•FNN gives the best performance for missile prediction.

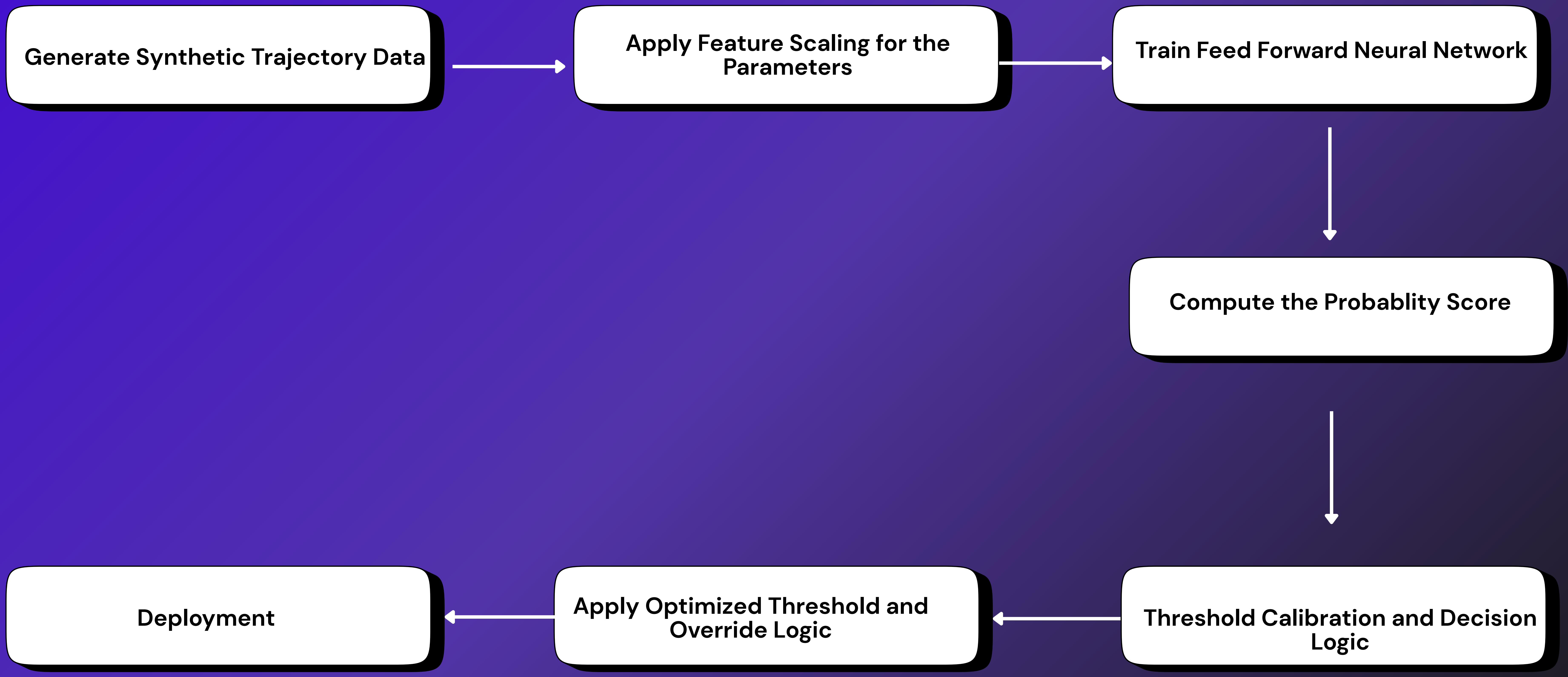
•Accuracy ≈ 99% with very fast inference (0.28 ms) → real-time ready.

•Logistic Regression & SVM = decent results; Decision Tree = weakest.

•Robustness tested by noise variation & feature removal.

•System is accurate, fast, and reliable for defense use.

WORKFLOW


Generate Synthetic Trajectory Data
Apply Feature Scaling for the Parameters
Train Feed Forward Neural Network
Compute the Probablity Score
Threshold Calibration and Decision Logic
Apply Optimized Threshold and Override Logic
Deployment

RESULT

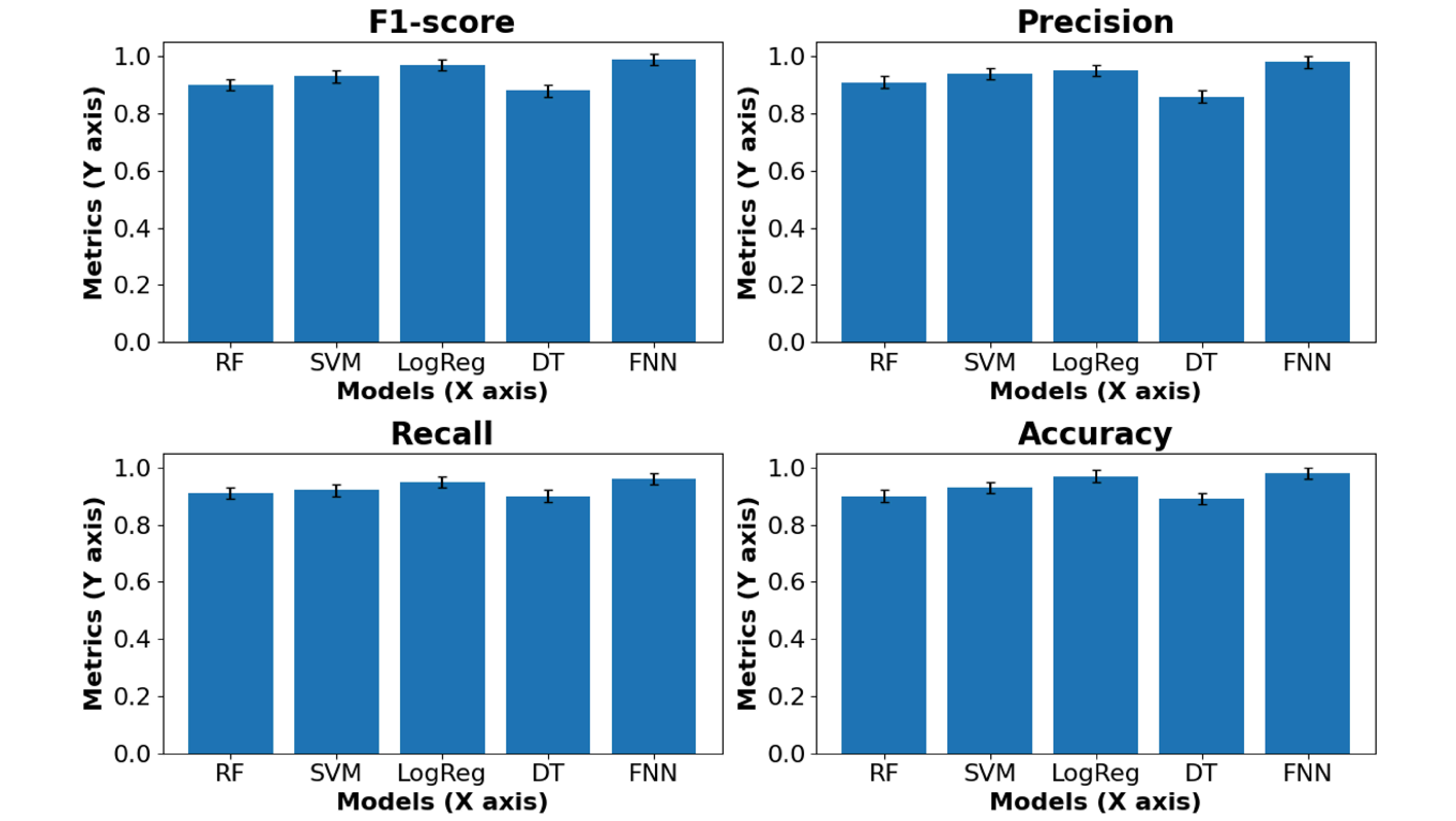
Model	Accuracy	F1-Score
Random Forest	88.60%	0.89
SVM	94.20%	0.94
Logistic Regression	97.50%	0.97
Decision Tree	86.20%	0.86
FNN (Proposed)	**99.00%**	**0.99**

Table 1. compares the performance of different models such as Random Forest, SVM, Logistic Regression, Decision Tree and FNN on the first dataset using raw input features.

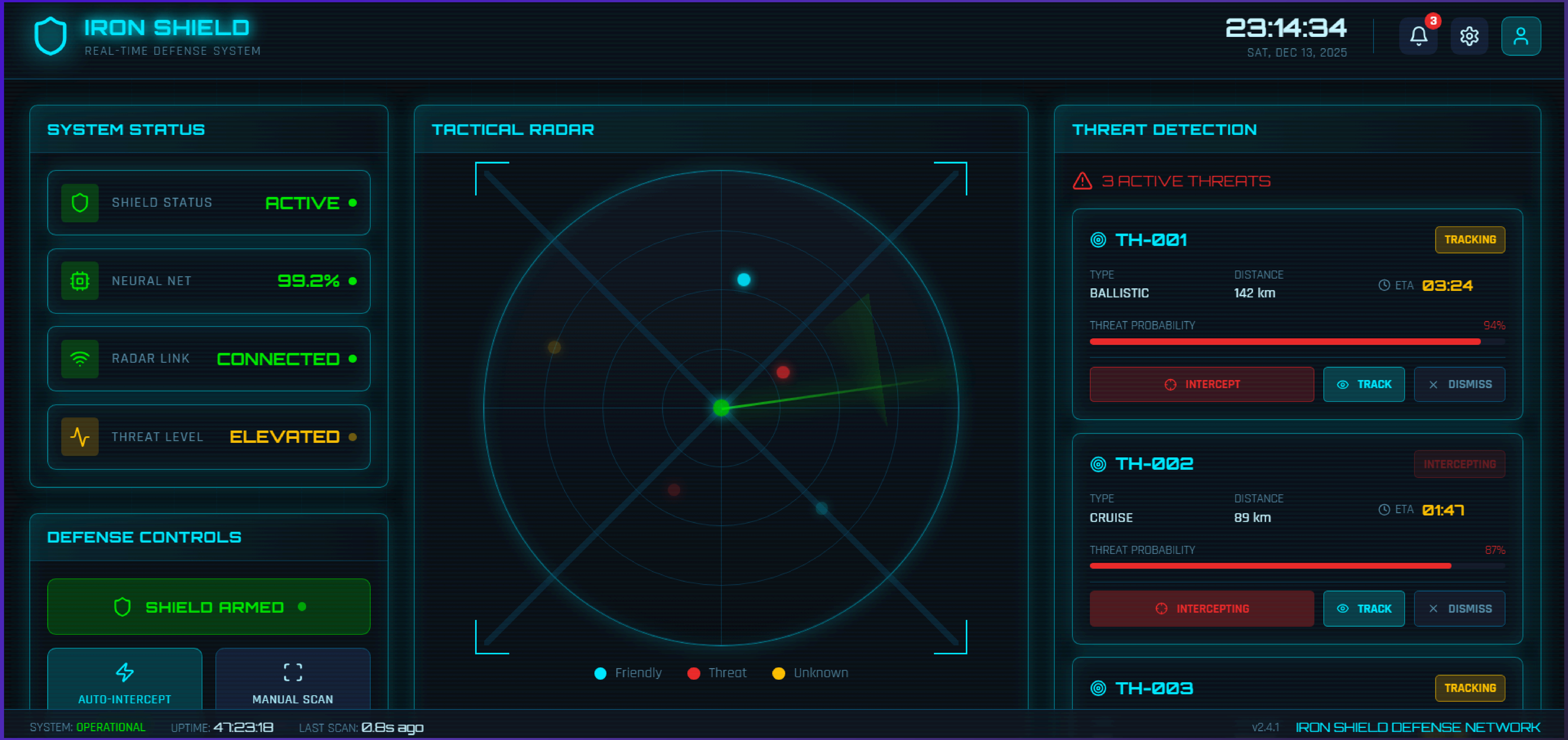
Model	Inference Time (ms)
Random Forest	2.14
SVM	3.5
Logistic Regression	3.12
Decision Tree	4.15
FNN (Proposed)	**0.28**

Table 2. Represents the different model with the inference time

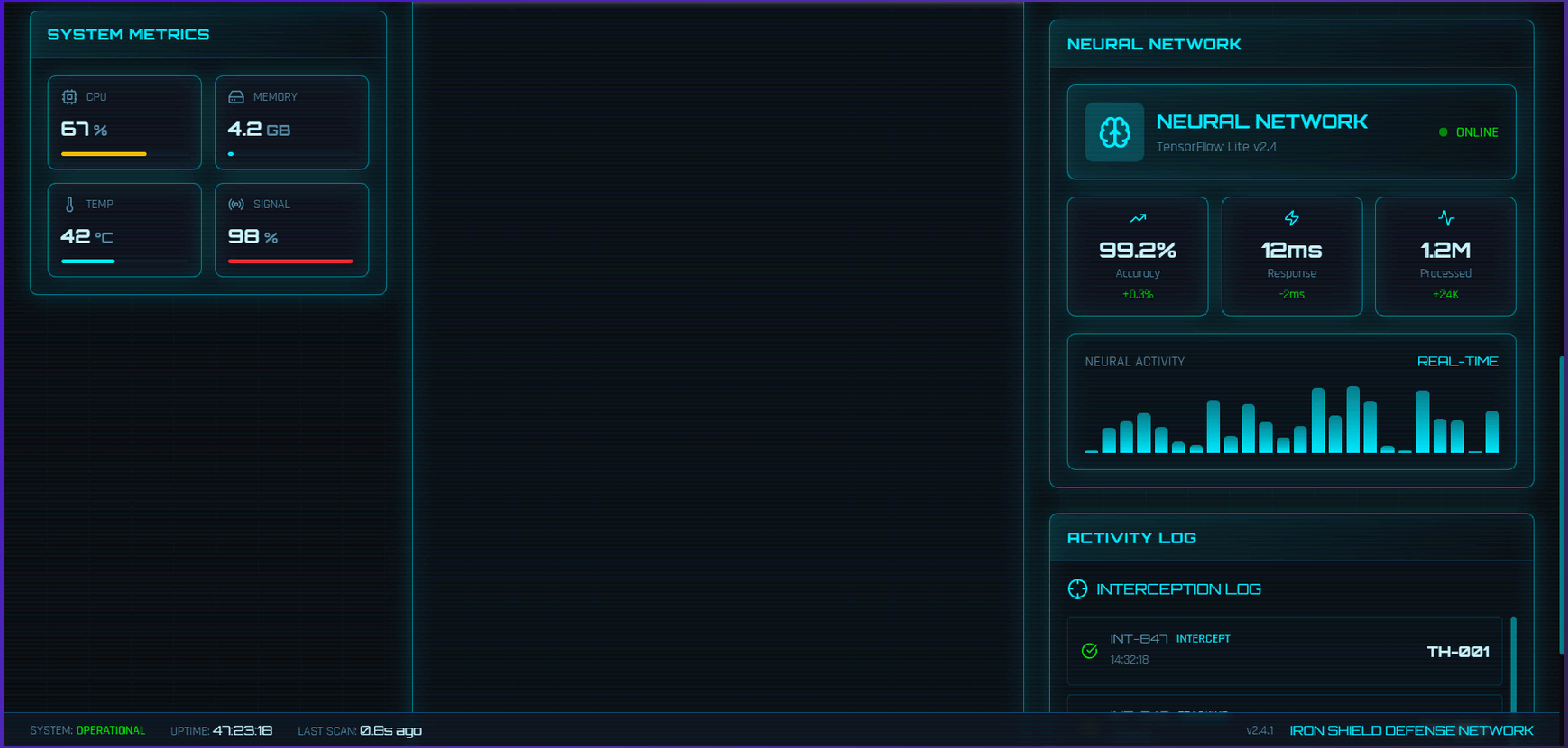
RESULT


F1-score
Precision
Recall
Accuracy
Metrics (Y axis)
Models (X axis)
1.0
0.8
0.6
0.4
0.2
0.0
RF
SVM
LogReg
DT
FNN

Simulated Real-Time Deployment Dashboard (Prototype)



Simulated real-time deployment interface demonstrating system behavior



Simulated real-time deployment interface demonstrating system behavior

Novelty

Physics-Guided AI Defense

1

Why Physics + AI?

Realistic trajectory learning with intelligence

2

Why Not Fixed Rules?

Learns and adapts dynamically

3

Why Safety Logic?

Prevents false interception

4

Why Edge Deployment?

Enables real-time decisions

Research Validation

- Paper Title:

  Iron Shield Real Time Defense System Using Neural Network

- CONFERENCE:

  IEEE ICETCI 2025

- Status:

  Published in IEEE Xplore

- Access:

  https://ieeexplore.ieee.org/document/11258004

Thank you

Project developed using a phase-wise prototype-to-deployment approach.