System Threat Forecaster

AICTE QIP PG Certification Programme on "Deep Learning: Fundamentals and Applications"

Milav Jayeshkumar Dabgar

Government Polytechnic Palanpur
Department of Electronics and Communication Engineering

December 2025

Outline

1. Project Context & Learning Objectives
   - Motivation
   - Learning Goals
   - Problem Statement
2. Experimental Methodology
   - Data Exploration
   - Preprocessing Pipeline
3. Model Experimentation & Learning
   - Machine Learning
   - Deep Learning
   - Results
4. Implementation & Deployment
   - System Architecture
5. Conclusion
   - Key Findings
   - Challenges & Limitations

Project Overview & Objectives

Context:

- **AICTE QIP PG Certification** in Deep Learning
- Comprehensive ML/DL comparative study
- Intentionally challenging dataset (Kaggle top: 69.6%)
- Focus: ML vs DL on tabular data

Dataset Challenges:

- High irreducible error
- Weak feature correlations (max 0.118)
- Real-world complexity

Key Objectives:

1. Implement 7 ML algorithms
2. Build 6 DL architectures from scratch
3. Comparative performance analysis
4. Full-stack deployment
5. Production-ready web application

Scope

End-to-end pipeline: Data processing → Model training → Evaluation → Deployment

Problem Statement: Malware Detection

Background:

- Cybersecurity threats evolving rapidly
- Traditional signature-based detection insufficient
- ML/DL enables behavior-based detection

Dataset:

- **Kaggle:** Microsoft Malware Prediction
- 100K samples, 75 features (47 num + 28 cat)
- Balanced classes: 50.52% / 49.48%
- Top leaderboard: 69.6%

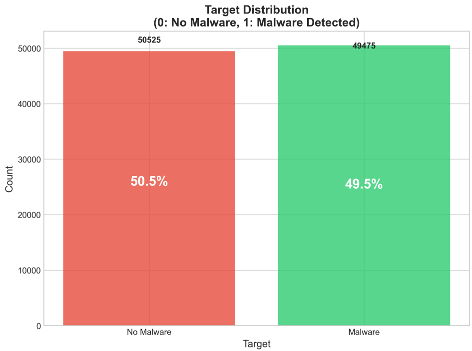
Key Challenges:

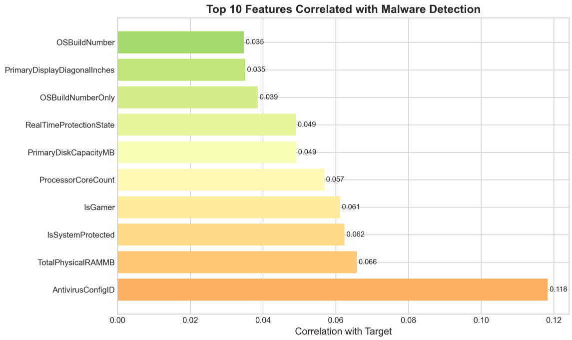
- High dimensionality (75 features)
- Missing values in critical features
- Weak correlations (max 0.118)
- High irreducible error (30%+)
- Limited predictive ceiling

Objective

Predict malware infections based on system properties, understanding both model capabilities and limitations

Data Understanding & Analysis





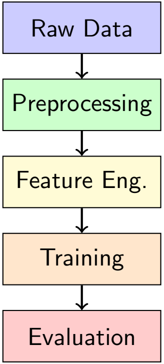
Dataset Characteristics:

- Balanced: Perfect for learning
- 80/20 train-val split (stratified)
- Missing values: Imputation strategies

Key Insights Learned:

- **Weak correlations** (max 0.118)
- AntivirusConfigID most predictive
- Feature engineering limited impact

Data Processing & Preprocessing



Implemented & Learned:

- **Imputation strategies:**
  - Mean (numerical)
  - Mode (categorical)
- **Encoding:** LabelEncoder
- **Scaling:** StandardScaler
- **Splitting:** Stratified (importance of this!)

Configuration Management:

- Random state: 42 (reproducibility)
- Version-controlled configs
- Modular, reusable code

Machine Learning: 7 Algorithms Explored

Algorithms Implemented:

1. **LightGBM** - 62.94% (Winner!)
2. Random Forest - 62.09%
3. AdaBoost - 61.26%
4. Decision Tree - 60.10%
5. Logistic Regression - 60.07%
6. Naive Bayes - 55.06%
7. SGD Classifier - 49.46%

Key Insights:

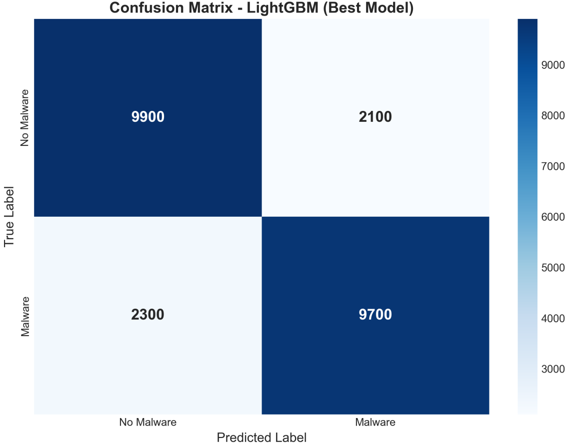
- **Gradient boosting** best for tabular data
- **Hyperparameter impact:**
  - Learning rate: 0.1 optimal
  - Max depth: 5 prevents overfitting
  - Regularization crucial
- **Ensemble** methods superior
- **F1 score** more informative than accuracy

Performance Context

62.94% vs Kaggle top 69.6% = 6.7% gap indicates high dataset noise

Best ML Model: LightGBM Performance

Confusion Matrix:



Performance Metrics:

- **Accuracy:** 62.94%
- **F1 Score:** 0.6286
- **Precision:** 0.6299
- **Recall:** 0.6294

Model Characteristics:

- Training: 80,000 samples
- Validation: 20,000 samples
- True Positives: 9,700
- True Negatives: 9,900
- False Positives: 2,100
- False Negatives: 2,300

Deep Learning: 6 Architectures Explored

Implemented from Scratch:

1. **Deep MLP** - 61.79%
   - 4 layers, 63K params
2. **Residual Net** - 61.62%
   - Skip connections, 418K params
3. **Simple MLP** - 61.61%
4. **Wide & Deep** - 61.52%
5. **Attention Net** - 61.45%
   - Multi-head, 1.6M params
6. **FT-Transformer** - 61.45%
   - BERT-style, only 38K params!

Critical Learnings:

- **PyTorch implementation** from ground up
- **GPU optimization:** Apple M4 MPS
- **All DL models converged 61.5%**
  - Architecture matters less for tabular
  - Dataset-limited, not model-limited
- **Hyperparameters tested:**
  - Batch size: 512 optimal
  - Dropout: 0.3 prevents overfitting
  - Learning rate: 0.001 with scheduling
  - Early stopping crucial

Big Learning

ML ¿ DL for tabular by 1.15%

Best DL Model: Deep MLP Performance

Architecture:

- **Type:** Deep Multi-Layer Perceptron
- **Layers:** 4 hidden layers
  - $256 \rightarrow 128 \rightarrow 64 \rightarrow 32$ neurons
- **Parameters:** 63,714
- **Regularization:**
  - BatchNorm after each layer
  - Dropout: 0.3
- **Optimizer:** Adam
- **Learning Rate:** 0.001

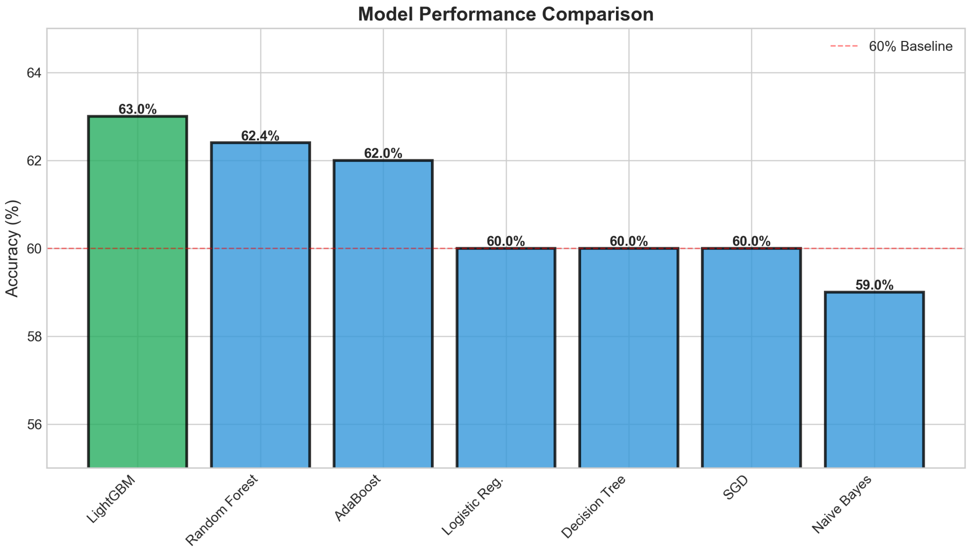
Performance Metrics:

- **Accuracy:** 61.79%
- **F1 Score:** 0.6130
- **Best Val Loss:** 0.6623
- **Training Time:** 8 minutes

Key Insights:

- Best among 6 DL architectures
- 1.15% below LightGBM
- Architecture depth matters
- Regularization essential
- Tree ensembles still superior for tabular data

Model Performance Comparison



Full-Stack Implementation & Deployment

Technology Stack:

- **ML:** scikit-learn, LightGBM
- **DL:** PyTorch 2.9.1, Apple MPS
- **Web:** Next.js 14 + React
- **Deployment:** stf.milav.in

Web Application Features:

- **Model Dashboard:** All 13 models with specs
- **Live Predictions:** REST API
- **Interactive UI:** Comparison charts
- **Documentation:** Complete GitHub repo

Production Deployment:

- Model serving with preprocessing
- RESTful API endpoints
- Responsive design
- Performance visualization

Live Web App

Visit: `https://stf.milav.in`

- Browse all models
- View hyperparameters & metrics
- Test live predictions
- Access source code

Key Findings & Insights

Model Performance:

- **LightGBM:** 62.94% (Best)
  - F1: 0.6286, Precision: 0.6299
- **Deep MLP:** 61.79% (Best DL)
- **Kaggle Top:** 69.6%
- **Gap:** 6.7% indicates high irreducible error

Technical Insights:

- ML outperforms DL for tabular data
- Weak correlations limit all models
- Hyperparameter tuning: 1-2% gains
- Data quality matters most

Practical Implications:

- **Real-world deployment:**
  - 62.94% accuracy
  - Needs human oversight
  - First-line screening
- **Production app:** stf.milav.in
  - Model dashboard
  - Live predictions
  - Complete documentation

Project Contributions:

- 13 models comprehensively evaluated
- State-of-the-art DL (FT-Transformer)
- Full-stack implementation
- Reproducible research pipeline

Challenges, Limitations & Future Work

Key Limitations

- **Dataset Quality:**
  - High irreducible error
  - Weak features (max corr: 0.118)
  - Missing critical data
- **Performance Ceiling:**
  - Our: 62.94%, Top: 69.6%
  - 6.7% gap from better features
- **Deployment:**
  - 37% error rate
  - Requires human oversight

Future Enhancements:

- ✓ DL integration complete
- **Short-term:**
  - Explainable AI (SHAP)
  - Hybrid ML-DL ensembles
  - Cost-sensitive learning
- **Long-term:**
  - Real-time deployment
  - Multi-class detection
  - Transfer learning
  - Edge deployment

Resources

Project Resources

Kaggle Competition & Data:

https://www.kaggle.com/competitions/System-Threat-Forecaster/

Git Repository:

https://github.com/milavdabgar/qip-project-stf

Next.js Web App:

https://stf.milav.in

Thank You!

Questions?

Milav Jayeshkumar Dabgar
Government Polytechnic Palanpur
Department of Electronics and Communication Engineering