Project ChurnBot: Full Strategic & Technical Report

Author: Phillip Harris **Tech Stack**: 🗄️ SQLite, 📊 Jupyter, 🐍 Python, 🔥 PyTorch, 💻 C++, 🔧 MLOps, 💻 TypeScript, 🐳 Docker, ⚛️ React, 🌐 Node.js

1. Executive Summary

ChurnBot is a domain-specific AI assistant for telecom churn prediction that demonstrates superior performance-interpretability trade-offs compared to general-purpose models. Unlike generic models, it detects telecom-specific behaviors—call patterns, service degradation, subscription anomalies—providing actionable insights and reducing churn-related losses through interpretable predictions.

Key Differentiators:

- Domain-specific cascade architecture achieves optimal balance of performance, interpretability, and computational efficiency
- Entirely local-first: no cloud, no external data transfer
- Dual interfaces: Terminal (light) & Dashboard (rich visualization)
- Modular architecture allows integration of IT and security monitoring pipelines later

2. Problem Statement

Traditional AI approaches often miss critical telecom-specific signals:

- Call patterns & usage anomalies
- Billing disputes & payment behavior
- Service degradation indicators
- Subscription plan changes

Impact: High false positives/negatives → wasted marketing spend, preventable churn, loss of revenue, and lack of actionable insights due to model opacity.

Solution: ChurnBot's interpretable three-stage cascade detects these patterns using specialized models with explainable decision paths: Random Forest → ANN → RNN.

3. Core Thesis (Refined)

Domain-Specific Cascade Architectures Achieve Superior Performance-Interpretability Trade-offs

Research Hypothesis: Domain-specific cascade architectures achieve superior performance-interpretability trade-offs compared to general-purpose models for specialized prediction tasks that can be decomposed into interpretable stages, demonstrated through telecom churn prediction.

Key Arguments:

- 🎯 **Architectural Interpretability**: Each cascade stage serves a distinct, interpretable purpose mapping to real telecom business logic
- ⚡ **Computational Efficiency Trade-offs**: Specialized models achieve comparable accuracy with dramatically lower resource requirements
- 🔍 **Domain Structure Exploitation**: Cascade design decomposes telecom churn into manageable, interpretable components
- 💡 **Actionable Insights**: Model predictions include clear feature importance and decision paths for business intervention
- 📊 **Measurable Explanations**: Quantifiable interpretability metrics enable comparison with black-box approaches

Scope Acknowledgment: This approach works best for domains where business processes can be decomposed into interpretable stages. Not claiming universal superiority across all problem types.

Churn Model Pipeline

Three-Stage Interpretable Cascade:

1. **Stage 1 – Lasso Logistic Regression (Feature Selection + Baseline Classifier)**

    - Elastic Net–style L1 penalty enforces sparsity, eliminating weak/redundant features.

    - Coefficients provide clear interpretability for business reasoning.

    - Balanced sampling pipeline applied before training.

2. **Stage 2 – Multi-Layer Perceptron (MLP Neural Network)**
    - Learns non-linear feature interactions beyond logistic regression.
    - Moderate architecture (100 → 50 neurons) with early stopping to prevent overfitting.
    - Provides interpretable layer-wise contribution analysis (limited neuron count).
3. **Stage 3 – Recurrent Neural Network (RNN)**
    - Custom PyTorch implementation (LSTM-based).
    - Treats features as ordered sequences to capture temporal/behavioral churn patterns.
    - Sequence dependencies interpreted via attention to later time steps in sequences.

Interpretability Framework

- **Lasso Logistic Regression (Stage 1):** Feature coefficients highlight the strongest churn drivers, directly interpretable by business stakeholders.
- **MLP (Stage 2):** Limited hidden layers allow partial interpretability of feature combinations and interactions.
- **RNN (Stage 3):** Final sequence modeling highlights temporal dynamics (e.g., tenure + payment behavior), with attention on changes toward churn.
- **Weighted Ensemble:** Final churn probability = 0.4 × Logistic + 0.3 × MLP + 0.3 × RNN (weighted toward more conservative logistic regression for stability).
- **Balanced Sampling:** Borderline-SMOTE + RandomUnderSampler ensure class balance for fairer training across all three models.

Pipeline Architecture:

data_loader → preprocessor → feature_engineer → leakage_monitor → cascade_model → experiment_runner → interpretability_analyzer

5. Empirical Validation Framework

Performance Metrics:

- **Traditional ML Comparison**: Precision, Recall, F1, PR-AUC vs. sklearn baselines
- **LLM Comparison**: Accuracy and efficiency vs. GPT-4/Claude on churn prediction tasks
- **Computational Efficiency**: Inference time, memory usage, energy consumption

Interpretability Metrics:

- **Feature Importance Clarity**: Quantified explanation quality across cascade stages
- **Decision Path Traceability**: Percentage of predictions with clear business rationale
- **Actionability Assessment**: Business user comprehension and intervention success rates

Generalization Testing:

- Cross-validation within telecom domain
- Temporal robustness across different time periods
- Dataset variation testing

6. IT & Security Pipelines

Goal: Demonstrate generalizability of cascade approach across enterprise domains while remaining local-first.

6.1 Anomaly / Intrusion Detection

- **Dataset**: flows.csv
- **Model**: Cascade approach adapted for security patterns

- **Interpretability**: Clear anomaly explanations for security analysts

6.2 Authentication / Account Abuse

- **Dataset**: auth_logs.csv
- **Model**: Temporal pattern recognition with explainable risk factors
- **Business Value**: Actionable security insights with clear reasoning

6.3 Ticket Classification & Routing

- **Dataset**: tickets.csv
- **Model**: Interpretable classification with routing rationale
- **Efficiency**: Fast local processing with explanation generation

7. C++ Optimization

Goal: Demonstrate computational efficiency advantages while maintaining interpretability.

Implementation:

- Custom RF, ANN, and RNN implementations optimized for telecom data patterns
- Interpretability-preserving optimizations (maintain decision path tracking)
- Performance benchmarking including explanation generation overhead

Optimization Techniques:

- Branch & bound algorithms
- Cache-friendly data structures for faster feature importance calculation
- SIMD matrix operations for ANN layers
- Custom memory allocators for temporal sequence processing

8. Privacy & Security Philosophy

- Local execution only (zero cloud dependencies)
- Complete data sovereignty for regulatory compliance
- Interpretable predictions reduce audit and compliance risks
- Optional API integrations require user-provided keys

9. Research Contribution Summary

Primary Contributions:

1. **Novel Architecture**: RF → ANN → RNN cascade for telecom churn with interpretability preservation
2. **Empirical Validation**: Comprehensive comparison against both traditional ML and modern LLMs
3. **Interpretability Framework**: Measurable explanation quality across cascade stages
4. **Efficiency Demonstration**: Performance-interpretability-efficiency trade-off analysis

Academic Positioning:

- Challenges "bigger is always better" assumption in current ML trends
- Provides concrete alternative to black-box model approaches
- Demonstrates practical value of domain-specific architectural design
- Contributes to interpretable AI research with quantifiable metrics

10. Roadmap & Research Milestones

Phase 1 (September-October 2025):

- Implement Python cascade with interpretability tracking
- Establish baseline performance and explanation quality metrics
- Begin C++ implementation for efficiency validation

Phase 2 (November 2025):

- Complete empirical validation framework
- Benchmark against traditional ML and LLM approaches
- Quantify interpretability advantages
- Draft research paper

Phase 3 (December 2025):

- Submit to academic conferences
- Extend validation to additional domains if time permits
- Refine for graduate school applications

11. Expected Research Impact

Academic Contributions:

- Evidence for domain-specific architectural advantages
- Quantifiable interpretability-performance trade-off analysis
- Challenge to current scaling paradigms in ML

Practical Benefits:

- Deployable enterprise solution with explainable predictions
- Reduced computational requirements for specialized tasks
- Clear business value through actionable insights

Scope Limitations:

- Results apply specifically to telecom churn and similar structured prediction tasks
- Interpretability benefits depend on domain decomposability
- Not claiming universal superiority over all model types