PROVISIONAL PATENT APPLICATION

Title: Adaptive Decision Trees for Quantum-Enhanced Cybersecurity Analysis with Real-Time Threat Routing

Inventor(s): Brian James Rutherford

Application Type: Provisional Patent Application

Filing Date: August 28, 2025

Application Number: [To be assigned by USPTO]

Inventors: [TO BE COMPLETED WITH ACTUAL INVENTOR NAMES]

Assignee: MWRASP Quantum Defense Systems, Inc.

Attorney Docket No: MWRASP-002-PROV

Filing Basis: 35 U.S.C. § 111(b) Provisional Application

TECHNICAL FIELD

The present invention relates to machine learning algorithms for cybersecurity applications, and more particularly to adaptive decision tree systems that optimize the routing of cyber threat analysis between quantum and classical computing resources based on real-time threat characteristics and system performance metrics.

BACKGROUND OF THE INVENTION

Current State of Cybersecurity Decision Systems

Modern cybersecurity systems rely on predetermined rules and static decision criteria to determine how cyber threats should be analyzed and processed. These systems typically use fixed algorithms that route all similar threat types through identical processing pipelines, regardless of current system capacity, threat urgency, or analysis complexity requirements.

Current cybersecurity decision systems suffer from several fundamental limitations:

1. Static Resource Allocation: Traditional systems use fixed routing rules that do not adapt to changing system conditions or threat landscapes.

2. Inefficient Resource Utilization: Without intelligent routing, systems either over-provision computational resources (leading to waste) or under-provision (leading to missed threats).

3. Lack of Learning Capability: Existing decision systems do not learn from historical performance to improve future routing decisions.

4. Single-Architecture Bias: Current systems are designed for purely classical computing architectures and cannot effectively utilize quantum computing resources.

Limitations of Current Machine Learning Approaches

While some cybersecurity systems incorporate machine learning for threat detection, these approaches have significant limitations when applied to resource allocation and threat routing:

Limited Adaptability: Current ML systems in cybersecurity focus on threat classification rather than optimal resource allocation for threat analysis.

Performance Blind Spots: Existing systems do not consider the computational efficiency and accuracy trade-offs between different processing approaches.

Quantum-Classical Integration Gap: No existing systems provide intelligent decision-making for hybrid quantum-classical cybersecurity architectures.

Need for Adaptive Decision Systems

The emergence of quantum computing in cybersecurity creates new opportunities for enhanced threat analysis, but also introduces complex resource allocation challenges:

1. Multi-Modal Processing: Different types of cyber threats benefit from different computational approaches (quantum vs. classical).

2. Dynamic Optimization: Optimal resource allocation changes based on current system load, threat characteristics, and available computational resources.

3. Real-Time Decision Requirements: Cybersecurity applications require microsecond-level decision making that traditional rule-based systems cannot provide.

4. Continuous Learning: The evolving threat landscape requires systems that continuously learn and adapt their decision-making strategies.

SUMMARY OF THE INVENTION

The present invention provides adaptive decision tree systems specifically designed for quantum-enhanced cybersecurity applications. The system implements machine learning algorithms that continuously optimize the routing of cyber threat analysis between quantum and classical computing resources based on real-time threat characteristics, system performance metrics, and historical analysis outcomes.

Primary Objectives

The adaptive decision tree system addresses the limitations of prior art by providing:

1. Intelligent Threat Routing: Advanced decision algorithms that analyze cyber threat characteristics and route analysis tasks to optimal computational resources.

2. Real-Time Adaptability: Decision trees that adapt in real-time based on current system capacity, threat urgency, and analysis requirements.

3. Continuous Learning: Machine learning algorithms that improve decision accuracy based on historical performance data and analysis outcomes.

4. Quantum-Classical Optimization: Specialized decision criteria optimized for hybrid quantum-classical cybersecurity architectures.

Key Technical Innovations

Adaptive Decision Tree Architecture: Novel tree structures that dynamically modify decision criteria and routing paths based on real-time system performance and threat analysis results.

Multi-Criteria Optimization Algorithms: Advanced optimization methods that consider multiple factors simultaneously: threat complexity, system capacity, analysis accuracy requirements, and processing time constraints.

Quantum-Aware Decision Nodes: Specialized decision tree nodes that understand quantum computing capabilities and limitations for different types of cybersecurity analysis tasks.

Continuous Learning Framework: Machine learning algorithms that continuously update decision tree parameters based on analysis outcomes and system performance metrics.

DETAILED DESCRIPTION OF THE INVENTION

System Architecture Overview

The adaptive decision tree system comprises several interconnected components that work together to provide optimal threat routing decisions:

1. Threat Characterization Engine

The threat characterization engine analyzes incoming cyber threats and extracts relevant features for decision-making:

1.1 Multi-Dimensional Feature Extraction

- Threat Type Classification: Identifies malware families, attack vectors, and threat categories

- Complexity Analysis: Evaluates computational complexity requirements for different analysis approaches

- Urgency Assessment: Determines time-critical nature of threats based on potential impact

- Data Volume Metrics: Measures data size and processing requirements

1.2 Quantum-Classical Suitability Analysis

- Quantum Advantage Assessment: Determines whether quantum algorithms provide computational advantages for specific threat types

- Classical Efficiency Evaluation: Identifies threats that are more efficiently processed using classical methods

- Hybrid Processing Identification: Recognizes threats that benefit from combined quantum-classical analysis

2. Adaptive Decision Tree Framework

The core decision tree system implements novel adaptive algorithms specifically designed for cybersecurity resource allocation:

2.1 Dynamic Tree Structure

```

Tree Node Structure:

- Node_ID: Unique identifier for decision node

- Decision_Criteria: Multi-dimensional criteria for routing decisions

- Quantum_Branch: Path for quantum processing allocation

- Classical_Branch: Path for classical processing allocation

- Hybrid_Branch: Path for combined processing allocation

- Performance_Metrics: Historical success rates for each branch

- Adaptation_Parameters: Learning rates and update mechanisms

```

2.2 Multi-Criteria Decision Nodes

Each decision node evaluates multiple criteria simultaneously:

```

Decision Criteria Framework:

1. Threat_Complexity_Score = complexity_analysis(threat_features)

2. Quantum_Advantage_Score = quantum_benefit_assessment(threat_type)

3. System_Capacity_Score = resource_availability_analysis()

4. Urgency_Score = time_criticality_assessment(threat_impact)

5. Historical_Performance_Score = lookup_success_rate(similar_threats)

Combined_Score = weighted_sum(all_scores, adaptive_weights)

```

2.3 Adaptive Weight Optimization

The system implements continuous learning algorithms that optimize decision criteria weights:

```

Algorithm: Adaptive Weight Optimization

Input: Historical_Results H, Current_Weights W, Learning_Rate α

Output: Updated_Weights W'

1. For each completed threat analysis:

a. Actual_Performance = measure_analysis_outcome()

b. Predicted_Performance = apply_decision_tree(threat_features, W)

c. Error = Actual_Performance - Predicted_Performance

2. Calculate weight gradients:

∂Error/∂W_i = gradient_calculation(Error, feature_contributions)

3. Update weights using gradient descent:

W'_i = W_i - α (∂Error/∂W_i)

4. Apply regularization to prevent overfitting:

W'_i = apply_regularization(W'_i, regularization_factor)

5. Validate updated weights against holdout dataset
```

6. Return W'

```

#### 3. Quantum-Classical Resource Optimization

The system implements specialized algorithms for optimizing resource allocation between quantum and classical processors:

**3.1 Quantum Resource Assessment**

- Coherence Time Evaluation: Assesses available quantum coherence time for analysis tasks

- Error Rate Analysis: Evaluates quantum processor error rates and their impact on analysis accuracy

- Queue Length Monitoring: Tracks quantum processor utilization and queue lengths

- Algorithm Compatibility: Determines compatibility between threats and available quantum algorithms

**3.2 Classical Resource Assessment**

- CPU Utilization Monitoring: Tracks classical processor availability and current workload

- Memory Availability Analysis: Evaluates available memory for different analysis approaches

- I/O Bandwidth Assessment: Measures network and storage bandwidth for data processing

- Parallel Processing Capacity: Assesses ability to parallelize classical analysis tasks

**3.3 Dynamic Resource Allocation Algorithm**

```

Algorithm: Dynamic Resource Allocation

Input: Threat T, Quantum_State Q, Classical_State C, Decision_Tree DT

Output: Allocation_Decision A

1. Extract threat features: F = extract_features(T)

2. Evaluate quantum suitability:

Q_suitability = quantum_advantage_score(F, Q.error_rate, Q.coherence_time)

3. Evaluate classical suitability:

C_suitability = classical_efficiency_score(F, C.cpu_available, C.memory_available)

4. Apply decision tree:

A = traverse_decision_tree(DT, F, Q_suitability, C_suitability)

5. Validate allocation against resource constraints:

If resource_available(A.allocation_type):

Return A

Else:

A = fallback_allocation(F, available_resources)

Return A

```

#### 4. Continuous Learning and Adaptation System

The system implements advanced machine learning algorithms that continuously improve decision-making performance:

**4.1 Performance Monitoring Framework**

- Analysis Accuracy Tracking: Monitors the accuracy of threat analysis results for different routing decisions

- Processing Time Measurement: Tracks processing times for quantum vs. classical analysis approaches

- Resource Utilization Metrics: Measures efficiency of resource utilization for different decision paths

- False Positive/Negative Analysis: Evaluates decision quality based on analysis outcomes

**4.2 Reinforcement Learning Integration**

The system implements reinforcement learning algorithms that optimize decision-making through reward-based learning:

```

Reinforcement Learning Framework:

State Space S: {threat_features, system_state, resource_availability}

Action Space A: {quantum_allocation, classical_allocation, hybrid_allocation}

Reward Function R: combination of (accuracy, speed, resource_efficiency)

Q-Learning Update Rule:

Q(s,a) = Q(s,a) + α[R(s,a) + γ max_a'(Q(s',a')) - Q(s,a)]

Where:

- α = learning rate

- γ = discount factor

- s' = next state after taking action a in state s

```

**4.3 Online Learning Adaptation**

The system continuously updates decision tree parameters based on streaming performance data:

```

Online Learning Algorithm:

For each new threat analysis result:

1. Update feature statistics: μ_new = update_mean(μ_old, new_result)

2. Update decision thresholds: threshold_new = adapt_threshold(accuracy_trend)

3. Modify tree structure if needed:

- Add new decision nodes for previously unseen threat patterns

- Prune nodes that consistently perform poorly

- Adjust branching criteria based on performance metrics

4. Validate changes against holdout dataset

5. Apply changes if validation successful

```

### Advanced Decision Tree Algorithms

#### Multi-Objective Optimization Decision Nodes
```

The system implements decision nodes that optimize multiple objectives simultaneously:

Objective Functions:

1. Accuracy Maximization: max(P(correct_classification | routing_decision))

2. Latency Minimization: min(processing_time | routing_decision)

3. Resource Efficiency: max(resource_utilization_efficiency | routing_decision)

4. Cost Optimization: min(computational_cost | routing_decision)

Pareto Optimization Approach:

```

Multi-Objective Decision Algorithm:

Input: Threat T, Objectives O = {accuracy, latency, efficiency, cost}

Output: Optimal_Route R

1. Generate candidate routing decisions: C = {quantum, classical, hybrid}

2. For each candidate c in C:

- Calculate objective scores: scores_c = evaluate_objectives(c, T, O)

3. Find Pareto optimal solutions:

Pareto_Set = {c | no other c' dominates c across all objectives}

4. If |Pareto_Set| == 1:

R = Pareto_Set[0]

Else:

R = select_from_pareto_set(Pareto_Set, current_priorities)

5. Return R

```

Hierarchical Decision Tree Architecture

The system implements hierarchical decision trees that operate at multiple levels of granularity:

Level 1: High-Level Routing Decisions

- Quantum vs. Classical vs. Hybrid allocation

- Resource priority assignment

- Urgency-based fast-track routing

Level 2: Algorithm-Specific Decisions

- Specific quantum algorithm selection (if quantum route chosen)

- Classical algorithm optimization (if classical route chosen)

- Hybrid processing coordination (if hybrid route chosen)

Level 3: Parameter Optimization

- Algorithm parameter tuning

- Resource allocation fine-tuning

- Quality-speed trade-off optimization

Real-Time Performance Optimization

Microsecond-Level Decision Making

The system is optimized for extremely low-latency decision making required for real-time cybersecurity applications:

Performance Optimizations:

1. Pre-computed Decision Paths: Common decision paths are pre-computed and cached for immediate access

2. Parallel Tree Evaluation: Multiple decision tree branches are evaluated in parallel using vectorized operations

3. Approximate Decision Making: For time-critical decisions, approximate algorithms provide rapid routing decisions

4. Predictive Resource Allocation: Machine learning models predict future resource needs and pre-allocate resources

Latency Optimization Algorithm:

```

Fast Decision Algorithm:

Input: Threat T, Time_Budget Δt, Decision_Tree DT

Output: Route R, Confidence C

1. If Δt < critical_threshold:
```

Use fast approximate decision

R = fast_approximate_route(T, cached_patterns)

C = estimate_confidence(R, T)

2. Else if Δt < standard_threshold:

Use optimized tree traversal

R = optimized_tree_traversal(DT, T, time_budget=Δt)

C = calculate_confidence(R, traversal_depth)

3. Else:

Use full decision tree with optimization

R = full_tree_evaluation(DT, T)

C = full_confidence_calculation(R, T, DT)

4. Return R, C

```

**CLAIMS**

### Independent Claims

Claim 1: An adaptive decision tree system for quantum-enhanced computational analysis comprising:

a) a computational characterization engine configured to analyze computational workloads and extract multi-dimensional features including task type, complexity, priority, and quantum-classical processing suitability;

b) an adaptive decision tree framework comprising decision nodes with multi-criteria evaluation capabilities that route computational tasks to optimal processing resources based on workload characteristics and real-time system conditions;

c) a quantum-classical resource optimization system configured to assess and allocate computational resources between quantum and classical processors based on current capacity, error rates, and processing requirements;
```

d) a continuous learning and adaptation system configured to monitor computational outcomes and continuously optimize decision tree parameters and routing decisions based on historical performance data;

e) wherein the adaptive decision tree framework implements real-time decision algorithms that optimize multiple objectives simultaneously including computational accuracy, processing latency, resource efficiency, and computational cost.

Claim 2: The adaptive decision tree system of Claim 1, wherein the decision nodes implement multi-criteria optimization algorithms that evaluate:

a) threat complexity scores based on computational requirements analysis;

b) quantum advantage scores based on the suitability of quantum algorithms for specific threat types;

c) system capacity scores based on real-time resource availability assessment;

d) urgency scores based on time-criticality and potential threat impact;

e) historical performance scores based on success rates of similar previous threat analyses.

Claim 3: The adaptive decision tree system of Claim 1, wherein the continuous learning and adaptation system implements:

a) reinforcement learning algorithms that optimize routing decisions through reward-based learning;

b) online learning algorithms that continuously update decision tree parameters based on streaming performance data;

c) adaptive weight optimization that modifies decision criteria importance based on analysis outcomes;

d) performance monitoring that tracks analysis accuracy, processing time, resource utilization, and decision quality metrics.

Dependent Claims

Claim 4: The adaptive decision tree system of Claim 1, wherein the decision tree framework implements hierarchical decision making at multiple granularity levels including high-level routing decisions, algorithm-specific selections, and parameter optimization.

Claim 5: The adaptive decision tree system of Claim 1, wherein the system achieves microsecond-level decision making through performance optimizations including pre-computed decision paths, parallel tree evaluation, and approximate decision algorithms.

Claim 6: The adaptive decision tree system of Claim 2, wherein the multi-criteria optimization implements Pareto optimization to identify optimal

routing decisions across multiple competing objectives.

Claim 7: The adaptive decision tree system of Claim 3, wherein the reinforcement learning algorithms implement Q-learning with state spaces representing threat features and system states, action spaces representing routing decisions, and reward functions combining accuracy, speed, and resource efficiency metrics.

Claim 8: A method for adaptive computational task routing in quantum-enhanced computing systems comprising:

a) characterizing incoming computational workloads by extracting multi-dimensional features including task type, complexity, priority, and processing suitability;

b) evaluating routing options using adaptive decision trees with multi-criteria decision nodes that assess quantum advantage, system capacity, and historical performance;

c) optimizing resource allocation between quantum and classical processors based on real-time capacity assessment and computational requirements;

d) executing computational analysis using allocated processing resources;

e) monitoring computational outcomes and updating decision tree parameters using continuous learning algorithms;

f) wherein the method achieves real-time routing decisions optimized for multiple objectives including accuracy, latency, efficiency, and cost across diverse computational applications.

Claim 9: The method of Claim 8, wherein the step of evaluating routing options implements Pareto optimization to identify optimal trade-offs between competing objectives.

Claim 10: The method of Claim 8, wherein the continuous learning implements both reinforcement learning for reward-based optimization and online learning for real-time parameter adaptation.

INDUSTRIAL APPLICABILITY

The adaptive decision tree system for quantum-enhanced cybersecurity analysis described herein has significant industrial applicability across multiple sectors requiring intelligent threat analysis and optimal resource allocation for cybersecurity operations.

Primary Industrial Applications

Enterprise Cybersecurity Operations Centers (SOCs): Large organizations can deploy this system to optimize the allocation of cybersecurity analysis resources between quantum and classical computing systems. The adaptive

decision trees enable SOCs to process higher volumes of threats with improved accuracy while minimizing computational costs and response times.

Managed Security Service Providers (MSSPs): Security service providers can utilize this technology to offer premium quantum-enhanced cybersecurity analysis services to their clients. The intelligent resource allocation ensures optimal utilization of expensive quantum computing resources while maintaining high-quality threat detection across multiple client environments.

Cloud Security Platforms: Major cloud providers can integrate these adaptive decision trees into their security platforms to provide intelligent threat routing services. The system's ability to make microsecond-level routing decisions makes it particularly valuable for real-time threat detection in high-throughput cloud environments.

Critical Infrastructure Protection: Power grids, telecommunications networks, and financial systems can implement this technology to ensure optimal allocation of cybersecurity analysis resources for protecting against sophisticated nation-state attacks that may employ both classical and quantum computing techniques.

Manufacturing and Commercial Deployment

Software Integration: The adaptive decision tree system is designed as software that can be deployed on existing hybrid quantum-classical computing infrastructure, making it immediately manufacturable and deployable without requiring specialized hardware development.

Scalable Architecture: The hierarchical decision tree structure allows for deployment across various scales, from single-organization cybersecurity systems to large multi-tenant security service platforms, ensuring broad commercial applicability.

API Integration: The system provides standardized APIs that enable integration with existing Security Information and Event Management (SIEM) systems, Security Orchestration, Automation and Response (SOAR) platforms, and other cybersecurity tools, facilitating easy adoption in enterprise environments.

Market Demand and Economic Impact

Resource Optimization Value: The system addresses the critical challenge of optimizing expensive quantum computing resources for cybersecurity applications. As quantum computing becomes more commercially available, the need for intelligent allocation systems becomes essential for cost-effective quantum-enhanced cybersecurity.

Performance Improvement: The continuous learning capabilities provide measurable improvements in threat detection accuracy and processing efficiency over time, delivering quantifiable return on investment for cybersecurity operations.

Competitive Advantage: Organizations deploying this technology gain competitive advantages through superior threat detection capabilities and

optimal resource utilization, addressing a multi-billion dollar cybersecurity market demand.

Technical Manufacturing Feasibility

The system leverages existing machine learning frameworks and quantum computing APIs, ensuring immediate technical feasibility for commercial deployment. The software-based nature of the invention eliminates complex manufacturing requirements while providing immediate value through intelligent automation of cybersecurity resource allocation decisions.

This invention solves real-world cybersecurity challenges by providing intelligent, adaptive resource allocation that cannot be achieved through static rule-based systems, making it immediately useful and commercially viable for industrial deployment across multiple cybersecurity market sectors.

DRAWINGS DESCRIPTION

Figure 1: Adaptive Decision Tree Architecture Overview

- Complete system architecture showing threat characterization engine, decision tree framework, resource optimization system, and learning components

- Data flow paths and control signals between system components

- Real-time feedback loops for continuous adaptation

Figure 2: Multi-Criteria Decision Node Structure

- Detailed view of decision node architecture with multiple evaluation criteria

- Quantum advantage assessment, system capacity analysis, and historical performance lookup

- Decision routing paths for quantum, classical, and hybrid processing

Figure 3: Continuous Learning Framework

- Reinforcement learning algorithm implementation with state spaces, action spaces, and reward functions

- Online learning adaptation mechanism for real-time parameter updates

- Performance monitoring and feedback integration

Figure 4: Hierarchical Decision Tree Structure

- Multi-level decision tree architecture with high-level routing, algorithm selection, and parameter optimization

- Decision granularity levels and their interconnections

- Performance optimization at each hierarchical level

Figure 5: Real-Time Performance Optimization

- Timeline diagram showing microsecond-level decision making processes

- Parallel processing and optimization techniques for low-latency decisions

- Performance metrics and optimization results

ABSTRACT

An adaptive decision tree system for quantum-enhanced cybersecurity analysis that intelligently routes cyber threats between quantum and classical computing resources. The system implements multi-criteria decision nodes that evaluate threat characteristics, system capacity, and historical performance to optimize routing decisions for accuracy, latency, efficiency, and cost. The system incorporates continuous learning through reinforcement learning and online adaptation algorithms that improve decision-making performance based on analysis outcomes. The adaptive decision tree framework operates at microsecond decision speeds while implementing hierarchical decision making across multiple granularity levels, providing optimal resource allocation for hybrid quantum-classical cybersecurity architectures.

INVENTOR DECLARATIONS

[TO BE COMPLETED WITH ACTUAL INVENTOR INFORMATION]

Primary Inventor: [Name]

- Contribution: Adaptive decision tree algorithms, multi-criteria optimization methods

Co-Inventor: [Name]

- Contribution: Continuous learning systems, reinforcement learning integration

Co-Inventor: [Name]

- Contribution: Real-time performance optimization, microsecond decision algorithms

ASSIGNEE INFORMATION

Assignee: MWRASP Quantum Defense Systems, Inc.

Address: [Company Address]

Assignment Date: [Date]

FILING INFORMATION

Application Type: Provisional Patent Application under 35 U.S.C. § 111(b)

Filing Date: August 25, 2025

Attorney Docket Number: MWRASP-002-PROV

Technology Center: 2100 (Computer Architecture and Software)

Art Unit: 2129 (Machine Learning and AI)

Document prepared: August 25, 2025

Filing priority: CRITICAL (within 1 week)

Status: READY FOR IMMEDIATE FILING