# AI MODEL RESEARCH REPORT

**Project Title: AI-Based Cyber Security Threats Prediction AI Agent**

## Categories of Models Used in Cybersecurity

| Category | Model Examples | Core Idea | Typical Use Case |
|---|---|---|---|
| Traditional Machine Learning Models | Decision Tree, Random Forest, SVM, KNN, Naïve Bayes | Learn statistical patterns in labeled data | Quick classification, small to medium datasets |
| Deep Learning Models | Artificial Neural Networks (ANN), CNN, RNN, LSTM | Learn complex temporal and spatial patterns | High accuracy, large datasets |
| Hybrid / Ensemble Models | Stacking, Boosting (XGBoost, AdaBoost), Voting Classifier | Combine multiple ML models to improve performance | Robust detection, better generalization |
| Reinforcement Learning Models | Deep Q-Networks (DQN), Actor-Critic | Learn through reward-based feedback | Adaptive threat response, AI agent behavior |

## Commonly Used Models in Research Papers

| Model | Type | Key Features | Accuracy (approx.) | Remarks |
|---|---|---|---|---|
| Random Forest (RF) | ML | Ensemble of Decision Trees | 97–99% | Highly stable and interpretable |
| Support Vector Machine (SVM) | ML | Finds hyperplane to separate classes | 95–98% | Good for smaller feature sets |
| K-Nearest Neighbors (KNN) | ML | Classifies based on neighboring samples | 90–95% | Simple, but computationally heavy |
| Naïve Bayes | ML | Probabilistic | 85–92% | Fast but less accurate for |

|  |  |  |  |  |
|---|---|---|---|---|
|  |  | classifier |  | complex data |
| Artificial Neural Network (ANN) | DL | Multi-layer perceptron learns complex mappings | 96–99% | Powerful but requires tuning |
| Convolutional Neural Network (CNN) | DL | Detects spatial patterns in features | 97–99% | Great for structured network data |
| Recurrent Neural Network (RNN/LSTM) | DL | Learns sequential or temporal dependencies | 98–99% | Excellent for time-based traffic data |
| XGBoost / LightGBM | Hybrid | Gradient boosting algorithms | 98–99% | Efficient, scalable, widely used |

## Analysis

The project aims to build an autonomous AI agent that monitors, predicts, and responds to threats. So, the model must be accurate, fast, capable of handling large, real-time data, generalizable to new attacks, and scalable for autonomous operation.

| **Model** | **Strengths** | **Weaknesses** | **Suitability** |
|---|---|---|---|
| Random Forest (RF) | High accuracy, robust, interpretable | Slower on very large data | Suitable for baseline |
| XGBoost | Extremely fast, efficient, and accurate | Complex tuning | Highly suitable |
| LSTM / RNN | Captures time-based traffic sequences | High computation cost | For advanced stage / real-time |
| CNN | Detects spatial patterns in flow features | Needs reshaping of input data | Optional for experiment |
| ANN | Strong generalization | Black-box behavior | For deep model version |

## Recommended Model Architecture

Based on project goals:
1. Start with Random Forest as the baseline model.
2. Move to advanced models like XGBoost for higher accuracy and efficiency.
3. Optionally integrate LSTM for temporal anomaly detection in future versions.
4. Hybrid approach: Combine Random Forest + XGBoost using Voting or Stacking Ensemble.

## Research Insights

Ensemble models (RF, XGBoost) consistently outperform single models in cybersecurity datasets. Deep models (LSTM, CNN) are best when real-time traffic patterns are needed. XGBoost offers a balance between accuracy, efficiency, and scalability, making it perfect for an AI agent framework.

## Final Model Selection

| Stage | Selected Model | Reason |
|---|---|---|
| Training & Testing | Random Forest | Excellent interpretability, easy to tune |
| Optimization | XGBoost | High accuracy, efficient with large data |
| Future Enhancement | LSTM-based Agent | Adds temporal pattern recognition for live detection |