AI/ML Models for Building an AI-Driven Cybersecurity Threat Prediction Platform

Cybersecurity threat prediction requires models that can:

- Detect **known attacks** (supervised learning)
- Identify **unknown / zero-day threats** (unsupervised learning)
- Understand **network behavior over time** (deep learning sequences)
- Interpret **logs, commands, and event descriptions** (NLP models)

The models below cover all 4 categories needed for a **modern enterprise-grade AI threat detection system**.

A. Machine Learning Models (Supervised)

These models are powerful for **structured tabular network data** such as NetFlow, packet statistics, system metrics, and file metadata.

1. Random Forest (RF)

Category: ML (Ensemble Trees)
Why this model matters in cybersecurity:
Random Forests are one of the strongest traditional ML baselines for intrusion detection because:

- They handle **high-dimensional traffic features** (duration, bytes, flags, ports).
- They automatically measure **feature importance** → useful for threat explanations.
- They are robust to noise, missing values, outliers — common in logs or network data.
- They scale extremely well and can be used in **real-time detection pipelines**.

Typical research accuracy:
Up to **99.9%** on NSL-KDD in multiple studies.

Use in our platform:
Perfect for initial threat scoring, real-time flow classification, and interpretable decision-making.

2. XGBoost

Category: ML (Boosting)
Why this model matters:
XGBoost is often the **top performer** on structured cybersecurity datasets:

- Excellent for **imbalanced datasets** (normal traffic >> attacks)
- Handles non-linear relationships
- Works extremely well on DDoS, brute force, and web attack datasets
- Efficient enough for large-scale SOC pipelines

Typical research accuracy:
Up to **99.91%** on CIC-IDS2017.

Use in our platform:
If you want a **highly accurate and stable attack prediction model**, XGBoost is one of the best supervised methods.

3. Support Vector Machine (SVM)

Category: ML (Kernel-based)
Why this model matters:
SVM is great when you have:

- Small-to-medium traffic representations
- Clean features
- Binary classification needs (benign vs malicious)

It performs well on **malware classification**, **host intrusion detection**, and **traffic behavior categorization**.

Typical accuracy: > 90% on NSL-KDD / IDS datasets.

Use in our platform:
Good for lightweight edge deployment or as a baseline to compare with more advanced models.

4. Logistic Regression (LR)

Category: ML (Linear Model)
Why this model matters:
While simple, Logistic Regression is very useful in cybersecurity:

- Fastest model to train → good for streaming data environments
- Useful for **attack probability scoring**
- Very interpretable (SOC analysts love this)
- Performs well with carefully engineered features

Typical accuracy: Around **85–92%** depending on dataset and preprocessing.

Use in our platform:
Best as a **baseline classifier** and for building interpretable threat scoring modules.

B. Deep Learning Models (Sequence- & Pattern-Aware)

These models learn patterns from **sequences, time-series traffic, raw packets, log streams**, etc.

5. RNN (Recurrent Neural Network)

Category: Deep Learning (Sequence Models)
Why this model matters:
Network behavior is **time-dependent**, and RNNs naturally capture this:

- Detect slow-developing attacks
- Learn behavior patterns over sessions
- Detect anomalies in login attempts, VPN activity, API calls
- Good for threat prediction over time windows

Typical accuracy: 80–90% depending on dataset size and sequence length.

Use in our platform:
Foundational sequence model to analyze user behavior analytics (UEBA), API abuse, and multi-step intrusion patterns.

6. LSTM (Long Short-Term Memory Networks)

Category: Deep Learning
Why this model matters:
LSTMs fix RNN weaknesses (vanishing gradients), making them extremely powerful for:

- **Time-series network traffic**
- **SSH authentication logs**
- **Wi-Fi handshake logs**
- **API call sequences**

LSTMs are particularly strong for **detecting persistent or multi-step threats**.

Typical accuracy: 88–95% on time-based IDS models.

Use in our platform:
Excellent for behavioral threat detection and anomaly prediction in time-based logs.

7. CNN-LSTM Hybrid

Category: Deep Learning (Hybrid)
Why this model matters:
Combines the strengths of:

- **CNN** → extracting localized patterns (packet bursts, byte sequences)
- **LSTM** → modeling long-range dependencies

Used extensively for:

- DDoS detection
- IoT intrusion detection
- Cloud attack detection
- Malware traffic pattern recognition

Typical accuracy: 95–97% on NSL-KDD, Bot-IoT, CIC datasets.

Use in our platform:
Ideal for multi-stage attack detection, where both micro-patterns and long-term behavior matter.

C. Unsupervised Anomaly Detection Models

These models detect **zero-day attacks**, which have no labeled examples.

8. Autoencoder (AE / Variational Autoencoder)

Category: Deep Learning (Unsupervised)
Why this model matters:
Autoencoders learn **normal traffic behavior** and flag deviations automatically:

- Detect zero-day attacks
- Great for anomaly detection in NetFlow and logs
- Low false-positive rate
- Works with both low and high-dimensional data

Typical performance: Strong anomaly detection with low FPR in research.

Use in our platform:
Best for unsupervised anomaly scoring modules.

9. Isolation Forest (IF)

Category: ML (Unsupervised)
Why it matters:
Simple but extremely effective for anomaly detection:

- No training labels required
- Fast on large datasets
- Works well for sparse attacks

- Very robust on web and IoT traffic

Typical research accuracy: Very strong performance on public anomaly datasets.

Use in our platform:
Recommended for **first-stage anomaly filtering** before advanced AI scoring.

D. NLP-Based Threat Detection Models (Logs & Commands)

Modern cybersecurity heavily relies on log analysis.

10. BERT (Transformer-based NLP model)

Category: AI / NLP
Why this model matters:
Logs, command-line traces, and error messages are **text**, not numbers.

BERT helps:

- Detect suspicious commands
- Identify malicious scripts (PowerShell, Bash, Python)
- Classify Windows/Linux event logs
- Detect social engineering signals
- Process descriptions of threat events

Typical accuracy: 90–96% on log anomaly datasets.

Use in our platform:
Perfect for the **log-analysis and SIEM augmentation** side of your threat prediction system.

📊 **COMPARISON TABLE:**

Model	Category	Supervised/ Unsupervised	Strengths	Weaknesses	Typical Accuracy
Random Forest	ML	Supervised	Interpretable, robust, fast	Needs feature engineering	Up to 99.9%
XGBoost	ML	Supervised	Best for tabular data, handles imbalance	Hyperparameter tuning needed	~99.9%
SVM	ML	Supervised	Strong for binary classification	Not ideal for very large datasets	>90%
Logistic Regression	ML	Supervised	Simple, explainable, fast	Limited to linear patterns	~85–92%
RNN	Deep Learning	Supervised	Learns sequences over time	Harder to train, slower	80–90%
LSTM	Deep Learning	Supervised	Excellent for time-series logs, avoids vanishing gradients	Requires compute, tuning	88–95%
CNN-LSTM	Deep Learning	Supervised	High accuracy, learns patterns + sequences	Heavy compute	95–97%
Autoencoder	Deep Learning	Unsupervised	Great for zero-day detection	Must tune reconstruction thresholds	Strong anomaly detection
Isolation Forest	ML	Unsupervised	Fast, scalable, no labels needed	Limited on highly structured data	Strong anomaly detection

BERT	NLP/ AI	Supervised/Unsupervised	Best for logs & commands, deep semantic understanding	Heavyweight model	90–96%

This model research discusses the comparative strengths, accuracy, and suitability of AI, ML, deep learning, anomaly detection, and NLP models for building a robust AI-driven cybersecurity threat prediction platform capable of detecting both known and zero-day attacks.