Model Research

Executive Overview

This report evaluates various machine learning and deep learning methodologies for constructing an AI-driven system to predict cyber threats. It assesses models based on key factors such as accuracy, response time, interpretability, data demands, robustness to changing environments, scalability, and deployment feasibility. The analysis provides guidelines on model selection tailored to specific use scenarios, including real-time threat detection, retrospective forensic examination, and risk assessment through graph-based methods. A stepwise implementation plan for model experimentation and eventual deployment is also outlined.

Research Goals & Performance Criteria

- Goals:
  - Anticipate cyber-attacks and detect anomalous activities across network and endpoint environments.
  - Deliver actionable risk ratings, alerts, and clear justifications to security operations center (SOC) teams.
- Metrics for evaluation:
  - Detection quality: Precision, Recall, F1 measure, and Area Under Precision-Recall Curve (AUC-PR), especially for skewed datasets.
  - Operational efficiency: Latency (milliseconds), event processing rate, false positive incidence, alert density per 1000 hosts.
  - Business impact: Average detection delay, reduction in analyst workload, financial implications of erroneous alerts.

Data Overview

- Sources: Network traffic records (NetFlow), endpoint logs (Syslog, EDR), access logs, firewall and IDS notifications, and threat intelligence feeds.
- Data characteristics: Predominantly imbalanced with rare malicious events, noisy, sequential/time-stamped records, and intricate relational data involving hosts, users, and IP addresses.
- Labeling challenges: Partial or weak annotations derived from alerts, expert triage, and sandbox analysis outcomes.

Explored Model Categories

1. Traditional Machine Learning Models

- Logistic Regression: Known for simplicity, transparency, and extremely low latency; performs best with well-crafted features and balanced input data.
- Ensemble Tree Methods (Random Forest, XGBoost, LightGBM): Popular benchmarks offering high accuracy on tabular data, resilient to scaling issues, moderately interpretable with tools like SHAP.
- Support Vector Machines (SVM), including One-Class SVM: Useful especially when negative samples are limited; scalability limitations with very large datasets.
- Isolation Forest and Local Outlier Factor (LOF): Unsupervised techniques proficient at detecting outliers without labels, suited for anomaly detection in structured feature spaces.

2. Deep Learning Architectures

- Multi-layer Perceptrons (MLP): Capture complex nonlinear relationships in features but require substantial training data and tuning.
- Recurrent Networks (LSTM, GRU): Designed for sequence modeling in user or host activity streams; good temporal pattern recognition albeit with higher inference cost.
- Transformer Networks: State-of-the-art for sequencing tasks with capacity to model long-range dependencies, though computationally intensive.
- (Variational) Autoencoders: Employed for unsupervised anomaly detection by reconstructing typical behavior patterns.
- Graph Neural Networks (GNNs): Leverage graph structures representing entities and their interactions to detect propagation and lateral movement paths in attacks; complexity in design and serving is higher.
- Contrastive and Self-supervised Approaches: Learn underlying representations from unlabeled data, enabling fine-tuning for specific threat prediction tasks.

Model Evaluation Summary

Model Type	Accuracy Potential	Latency	Data Needs	Interpretability	Stability to Drift	Deployment Complexity	Ideal Application
Logistic Regression	Low to Moderate	Very Low	Low	High	Low to Moderate	Low	Quick baseline, early detection
Random Forest / XGBoost	Moderate to High	Low to Medium	Medium	Medium (SHAP)	Medium	High	Structured data alert scoring

Model Type	Accuracy Potential	Latency	Data Needs	Interpretability	Stability to Drift	Deployment Complexity	Ideal Application
Isolation Forest / LOF	Moderate	Low	Low	Medium	Low to Medium	High	Unsupervised anomaly detection
SVM / One-Class SVM	Moderate	Medium	Low	Low	Low	Medium	Small dataset novelty detection
MLP	Moderate to High	Medium	Medium to High	Low	Medium	Medium	Capturing nonlinear feature interactions
LSTM / GRU	High	Medium to High	High	Low	Medium	Medium	Sequential event detection
Transformer-based Models	Very High	High	Very High	Low	Medium to High	High	Long-sequence modeling
Autoencoders / VAE	Moderate to High	Medium	High	Low	Low to Medium	Medium	Unsupervised anomaly spotting
GNNs	Very High	Medium to High	High	High	Low to Medium	High	Graph-based risk and lateral movement
Contrastive / Self-supervised	High (with tuning)	Medium to High	High	Low (pre-training)	High	Medium	Few-shot learning, feature extraction

Recommended Model Selection by Use Case

- Real-time low-latency detection: Favor lightweight models such as Logistic Regression or gradient boosted trees (LightGBM/XGBoost) optimized for fast inference and interpretability.
- Sequential user or host behavior analysis: Use LSTM, GRU, or Transformer models to capture temporal dependencies and patterns missed by feature engineering alone; consider hybrid solutions for latency-sensitive cases.
- Graph-centric threat scoring: Deploy GNNs to effectively evaluate relational interactions and multi-hop attack vectors in network graphs.
- Rare or emerging threat detection without labels: Utilize unsupervised detection via Autoencoders, Isolation Forest, or One-Class SVM, possibly integrated with supervised models in hybrid ensembles.

Practical Strategy

A multi-model ensemble approach combining fast scored models, sequence analysis, graph reasoning, and anomaly detection provides the best trade-off between accuracy, interpretability, and latency. The decision system can weigh models based on context and recent performance, optimizing for operational requirements.

Implementation Roadmap

1. Data ingestion and feature engineering pipeline setup.
2. Develop and benchmark classical models (Logistic Regression, XGBoost).
3. Train and evaluate unsupervised anomaly detectors.
4. Introduce sequence models (LSTM/Transformer) and graph-based models (GNN).
5. Build ensemble fusion logic and conduct tests in shadow mode.
6. Optimize model performance, deploy at scale, and implement monitoring for drift and degradation.

Best Practices

- Emphasize robust feature engineering and baseline modeling.
- Use time-aware validation and address class imbalance rigorously.
- Maintain rigorous version control and monitoring post-deployment.
- Incorporate human analysts for feedback loops and cautious automation policies initially.

Key References

- MITRE ATT&CK for threat technique mappings.
- Research on Transformer and Graph Neural Network applications in cybersecurity.
- Tools and libraries such as Feast, ONNX, Triton for scalable model deployment.

This version retains all major points and structure but uses original phrasing to ensure uniqueness. If desired, it can be further tailored to your specific project domain or focus areas.