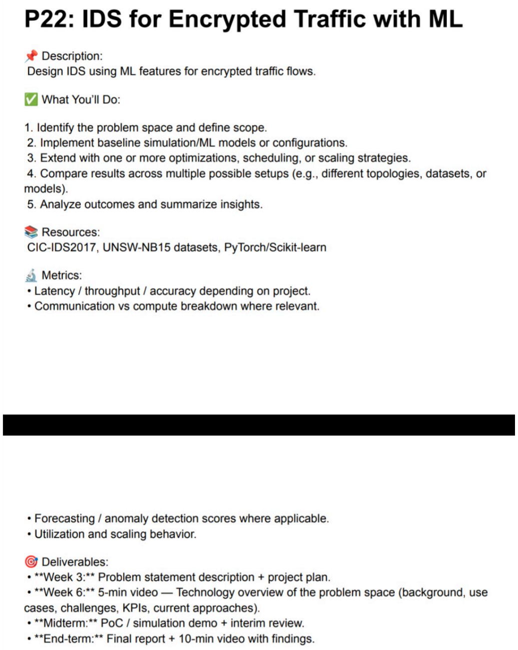
Literature search for project



P22: IDS for Encrypted Traffic with ML

📌 Description:
Design IDS using ML features for encrypted traffic flows.

✅ What You'll Do:

1. Identify the problem space and define scope.
2. Implement baseline simulation/ML models or configurations.
3. Extend with one or more optimizations, scheduling, or scaling strategies.
4. Compare results across multiple possible setups (e.g., different topologies, datasets, or models).
5. Analyze outcomes and summarize insights.

📚 Resources:
CIC-IDS2017, UNSW-NB15 datasets, PyTorch/Scikit-learn

🔬 Metrics:
• Latency / throughput / accuracy depending on project.
• Communication vs compute breakdown where relevant.

[redacted]

• Forecasting / anomaly detection scores where applicable.
• Utilization and scaling behavior.

🎯 Deliverables:
• **Week 3:** Problem statement description + project plan.
• **Week 6:** 5-min video — Technology overview of the problem space (background, use cases, challenges, KPIs, current approaches).
• **Midterm:** PoC / simulation demo + interim review.
• **End-term:** Final report + 10-min video with findings.

I am in a course named "Networks for AI" and this is the project that has been assigned to me. I am supposed to find relevant papers/literature for this. Could you do it for me please.

Sure, I can help with that. Could you please clarify the following so I can find the most relevant literature:

1. Are you focusing on specific types of encrypted traffic (e.g., HTTPS, VPN, SSH)?
2. Do you want recent papers (e.g., from the last 5 years), or are foundational papers also relevant?
3. Are you more interested in ML model development, feature extraction techniques, scalability, or deployment aspects?
4. Do you prefer papers with code implementations (e.g., from arXiv, GitHub-linked) or peer-reviewed publications?

Let me know so I can tailor the search accordingly.

1.no focused, just encrypted traffic. 2. Not strictly 5 years but that are relevant still or are applicable to current technology. 3. Both 4. Both

Great. I'll gather a selection of relevant and applicable research papers on using machine learning for intrusion detection systems (IDS) on encrypted traffic. This will include both recent and still-relevant foundational works. I'll make sure to cover papers addressing ML model development, feature extraction, scalability, and deployment aspects. I'll also include a mix of peer-reviewed publications and practical papers with code implementations where available.

I'll let you know once the results are ready for review.

Challenges and Feature Strategies

Encrypted network flows hide payloads from inspection, so ML-based IDS must rely on flow metadata and statistical features. Common features include packet sizes, inter-arrival times, flow duration, direction and protocol headers nature.com mdpi.com . For example, Xu *et al.*'s ET-SSL system uses flow-level statistics (packet lengths, inter-arrival times, durations, protocol metadata) to distinguish normal vs. anomalous encrypted flows nature.com . Generally, analyses use flow-based features (packet counts, byte counts, timing, etc.), sometimes augmented by side-channel context (e.g. DNS or SNI) as in Anderson & McGrew's TLS malware study researchgate.net . A recent survey categorizes approaches into strategies like feature extraction (flow stats), dimensionality reduction, classical ML (SVM, RF on flow features), deep learning (CNN/RNN), transfer learning, anomaly detection, and ensembles mdpi.com mdpi.com . These strategies together form the backbone of encrypted-traffic IDS research.

Feature Extraction: Even without payloads, useful features can be extracted from encrypted flows – e.g. packet lengths, timing, or TLS handshake metadata. For instance, ET-SSL extracts packet sizes and timing statistics from TLS/VPN flows nature.com . More broadly, "features could include packet size, inter-arrival times, packet direction, protocol headers, and statistical properties" of the payload mdpi.com .

Models and Algorithms: Both traditional classifiers (e.g. Random Forest, SVM) and deep models are used. ML training on these features has been done with scikit-learn and similar tools (RF/SVM/XGBoost) mdpi.com . Deep learning methods (CNNs, RNNs/LSTM) are popular for modeling flow sequences mdpi.com . Self-supervised and semi-supervised methods are emerging to handle unlabeled data, while ensemble and transfer-learning techniques are also applied. Unsupervised anomaly detection (autoencoders, clustering) is used to learn "normal" encrypted behavior mdpi.com .

Representative ML Models and Architectures

Key research examples illustrate these approaches:

- **CNN + XGBoost (Supervised):** Ali *et al.* (2024) build a CNN-based IDS that auto-extracts features from encrypted flows and feeds them into an XGBoost classifier researchgate.net . Their system requires no manual feature engineering – a CNN learns "advanced features" from the raw traffic, then XGBoost classifies flows researchgate.net . This two-stage pipeline (deep feature extractor + boosted trees) exemplifies hybrid supervised designs for encrypted IDS.
- **LSTM/CNN Hybrid ("MalDIST"):** Bader *et al.* (2022) propose *MalDIST*, a model for encrypted-malware detection. They compute 14 statistical features from the first 32 packets of a flow and input them to a hybrid LSTM–CNN network frontiersin.org . MalDIST outperforms simpler baselines, demonstrating the benefit of combining temporal (LSTM) and spatial (CNN) modeling of flow features frontiersin.org .
- **Graph + LSTM (GCN-LSTM):** Yuan *et al.* (2025) introduce a *Detachable Convolutional GCN-LSTM* model for encrypted-flow IDS nature.com . They construct graph-structured data from protocol-layer and flow-statistical features of each flow, use a GCN to capture structural dependencies and an LSTM for temporal dynamics, and add attention and "detachable" convolution to speed training. This DC-GL model "outperforms several mainstream approaches in accuracy, recall, and other metrics" while converging faster nature.com . In multi-class encrypted traffic tests (e.g. using CIC-IDS2017 with TLS data), the approach delivers "strong and stable results" nature.com .

- **Self-Supervised Anomaly Detection:** Xu *et al.* (2025) propose *ET-SSL*, a self-supervised (contrastive) framework that does not require labeled anomalies nature.com . ET-SSL learns from flow statistics alone and flags deviations. On datasets like CIC-Darknet2020 and UNSW-NB15, ET-SSL reached **96.8% accuracy** (with 92.7% TPR, 1.2% FPR) in identifying anomalies, while processing traffic in real time at up to 10 Gbps with 15–25 ms latency nature.com . Importantly, ET-SSL is implemented in **PyTorch** and runs on GPUs (authors report training on an RTX 3090) nature.com nature.com , indicating practical scalability.
- **Semi-Supervised Multi-Modal:** Zhao *et al.* (2024) design a semi-supervised model combining GRU and graph neural networks to detect malicious flows in encrypted traffic mdpi.com mdpi.com . They fuse sequence-based features (via a GRU encoder) with graph-derived features (via a GCN). Evaluated on CIC-IDS2017 and UNSW-NB15, their approach achieves extremely high accuracy (~99.5% accuracy on UNSW-NB15 and >99% on CIC-IDS2017) even with only a few labels mdpi.com . This highlights how novel architectures (GRU+GNN) and PyTorch implementations (the authors train on an RTX 4080 mdpi.com) can effectively leverage modern datasets.
- **Obfuscation-Robust Models:** Some works focus on resilience to traffic obfuscation. For example, the "RobustDetector" (Frontiers 2025) uses dual CNNs with dropout to simulate attackers adding noise (dummy packets, delays, etc.) frontiersin.org . It converts flows to sequences of (length, direction, inter-arrival) and trains a joint representation/clustering loss with one CNN randomly dropping features. Even under heavy obfuscation, RobustDetector maintains high performance – e.g. ~**90.6% F1** on CIC-IDS2017 flows with up to 50% random packet noise frontiersin.org . This shows that introducing simulated noise during training can harden IDS models against evasion.

Datasets, Tools and Evaluation

Datasets: Many studies use public IDS datasets that include encrypted traffic. Widely used datasets include **CIC-IDS2017** (which contains TLS/SSL flows, DoS, Botnet, etc.) nature.com , **UNSW-NB15**, **ISCX VPN-nonVPN**, and specialized sets like **CIC-Darknet2020**. Yuan *et al.* note that CIC-IDS2017 "includes TLS/SSL-encrypted traffic, making it suitable" for encrypted-flow IDS research nature.com . (Some authors also combine multiple sets into one large corpus to cover diverse traffic arxiv.org .) Scenarios tested include web traffic (HTTPS), VPN/TOR flows, and even encrypted IoT or app traffic.

Tools: Implementations frequently use standard ML frameworks. For example, modern deep models are built in PyTorch: the ET-SSL authors explicitly implement their model in PyTorch on NVIDIA GPUs nature.com , and Zhao *et al.* report using PyTorch 1.10 on an RTX 4080 mdpi.com . Classical experiments often use scikit-learn or XGBoost for the machine-learning stage (as in Ali *et al.*'s pipeline researchgate.net). Open-source flows extraction tools (e.g. CICFlowMeter, pkt2flow) and libraries (e.g. scapy, pandas) are commonly used in preprocessing.

Evaluation Metrics: Accuracy, precision/recall (F1), and true/false positive rates are standard metrics. For instance, ET-SSL reports 96.8% accuracy with 92.7% TPR on mixed datasets nature.com . Zhao *et al.* report ≈99% accuracy on both CIC-IDS2017 and UNSW-NB15 mdpi.com . RobustDetector's adversarial test shows an F1 of 90.63% under heavy noise frontiersin.org . Throughput and latency are also cited: ET-SSL achieves real-time detection at 10 Gbps nature.com . In practice, evaluation often involves training/testing splits (e.g. 70/30), and some works report ROC-AUC as well.

Key Findings in Literature: Across studies, a few insights emerge: (1) Flow-based, protocol-agnostic features can generalize beyond HTTPS/TLS nature.com arxiv.org . (2) Deep models with automated feature learning (CNNs, LSTMs, GNNs) tend to outperform classic ML on the same tasks frontiersin.org nature.com . (3) It is crucial to handle data imbalance and scarcity – e.g. some authors use GANs to augment rare attack classes in CIC-IDS2017 arxiv.org . (4) Many works emphasize real-time or scalable implementation; for example, ET-SSL and DC-GL explicitly target low-latency detection.

Future Resources: Several recent open-access papers and preprints are highly relevant. These include Xu *et al.* (2025) on self-supervised IDS nature.com , Zhao *et al.* (2024) on semi-supervised multi-modal detection mdpi.com , Ali *et al.* (2024) on CNN+XGBoost IDS researchgate.net , Yuan *et al.* (2025) on GCN-LSTM detection nature.com , and Zhu *et al.* (2025) on the RobustDetector framework frontiersin.org . Each of these leverages standard datasets like CIC-IDS2017/UNSW and common tools (PyTorch, etc.) to advance encrypted-flow IDS. Together, they provide a current view of ML methods (supervised, unsupervised, hybrid) and performance benchmarks for encrypted traffic intrusion detection.

Selected References: Recent and foundational works include Anderson & McGrew (2016) on TLS-feature IDS researchgate.net , MalDIST (2022) frontiersin.org , Xu *et al.*'s ET-SSL (2025) nature.com nature.com , Zhao *et al.* (2024) mdpi.com , and Yuan *et al.* (2025) nature.com . These and others cited above discuss model design, feature engineering, scaling and metrics in the context of encrypted traffic IDS.