International Journal of Artificial Intelligence, Data Science, and Machine Learning
Grace Horizon Publication | Volume 7, Issue 2, 70-77, 2026
ISSN: 3050-9262 | https://doi.org/10.63282/3050-9262.IJAIDSML-V7I2P111

Original Article

False Positive & False Negative Mitigation in ML-Based Threat Detection

John Komarthi
San Jose, CA.

Received On: 06/03/2026 ***Revised On: 30/03/2026*** ***Accepted On: 07/04/2026*** ***Published On: 18/04/2026***

Abstract - *Machine learning has become an important part of cybersecurity for detecting malware, fraud, intrusions, phishing, and multiple other threats. The ML systems also face challenges with false alarms (false positives) and missing detections (false negatives), which can significantly impact the effectiveness of the Security Operations Centers (SOCs). In this paper, a detailed analysis is done of the challenges that are associated with false positives and false negatives in ML-based threat detection, while exploring the root causes and the possible mitigation strategies. The data quality issues will be examined, such as the label noise, rare event imbalance, and evolving attack patterns. Also, several model-level strategies, which include the probability calibration, cost-sensitive learning, anomaly detection methods, threshold tuning, monitoring models in production, uncertainty estimation, and adversarial robustness. The operational best practices will also be discussed, which include the evolution metrics, creating feedback loops with analysts, monitoring models in production, integrating incident response processes, and incorporating human oversight. These practices ensure that robust deployment of machine learning systems, real-world examples from the industry, such as Intrusion Detection Systems (IDS), SOCs, email phishing detection, and fraud detection, illuminating the trade-offs that are involved and the lessons learned from various implementations. Furthermore, limitations will be addressed, ethical and regulatory concerns, and the potential ways in which the attackers might exploit the mitigation of the false positives and false negatives. This outlines the various mitigation methods, which highlight the trade-offs which are related to complexity, data requirements, and the typical impact on false positives and false negatives. Recommendations are offered, such as adapting to the multi-layered systems, fostering continuous learning, and interfacing explainable AI (XAI) approaches along with future research directions. This report aims to serve as a guide for security stakeholders dealing with ML-driven false positives and false negatives.*

Keywords - *False Positive, Machine Learning, Class Imbalance, Cost-Sensitive Learning, Uncertainty, Intrusion Detection, Fraud Detection, False Negative, Anomaly Detection, Model Calibration, Explainability, SOC Operations, Phishing, Monitoring.*

1. Introduction

False positives (FPs) and false negatives (FNs) in machine learning (ML) threat detection critically impact the security operations. High FP rates overwhelm analysts with the noise, which causes alert fatigue, and the attacks can be missed, while the false negatives leave the real threats undetected [1]. Cyber defenders are increasingly deploying ML models for identifying threats in large-scale enterprise data (host logs, network traffic, transactions, emails, etc.). Unlike the traditional signature rules, ML can recognize the complex patterns and novel attacks [2]. Every detection system has to balance two error types: false positives that are benign events flagged as malicious, and false negatives that are malicious events missed. High FP rates inundate analysts with noise, waste time, and increase the risk of alert fatigue. In the same way, false negatives let the threats slip through cracks, but they have a greater cost, which is breaches. For instance, a rule-based IDS can issue many alarms for benign maintenance traffic (FPs), and it still misses a new attack variant (FN). The phishing filter can be mislabeled as a legitimate email (FP) while letting a well-crafted spear-phish pass (FN). An ideal detection system minimizes both the FNs and FPs, but when it comes to practice, reducing the FPs tends to increase FNs and vice versa.

ML is applied across multiple security domains such as Network Intrusion Detection (NIDS), Endpoint Detection and Response (EDR), SIEM correlation, User/Entity Behavioral Analytics (UEBA), email/spam/phishing filter, fraud detection in finance, etc. The traditional security relies heavily on the signatures or the heuristics, for example, a known malware hash or a Snort rule. These are precise for known threats but fail on unknown (zero-day) attacks. The ML supervised classifiers, anomaly detectors, and deep models promise to generalize to novel patterns. A trained model can generally flag an unusual process activity or network flows without explicit rules. The unsupervised techniques build a model of normal to detect the outliers. However, the real deployments show the pitfalls. A widely cited study found that the anomaly-based IDS “have shown great promise to detect the previously unseen threats, but their success has been limited, due to the excessive number of false positives that they produce” [3]. The ML models do make errors, but a research review notes that the standard ML classifiers give output with high confidence even for misclassified or out-of-distribution inputs [4]. The deployed systems combine the ML with rule-based or manual review layers. In practice, the defenders have to tune the models to hit an acceptable precision-recall balance, given their tolerance for risk and resource constraints. Recent research shows the urgency: “IDSs have been experiencing significant losses in detection and effectiveness,” motivating a deep analysis of false negatives [5].

The signature-based tools often suffer from the inability to detect zero-day attacks [2], while the anomaly detectors have encountered many obstacles and do not produce reliable results, which limits the widespread adoption. The combination of evolving threats and big data means the ML performance degrades over time if ignored and needs careful maintenance.

2. Causes of False Positives and False Negatives

False Positives (FPs) happen when benign events are flagged as threats; the common reasons for this include overly general rules or features, poor data representativeness, and lack of context. For instance, an IDS alerts on large file transfers, yet that can be routine backups. Corelight explains this as, FPs are a symptom of a deeper problem of poorly tuned tools that lack the necessary context and data fidelity [6]. When the model is trained on stale or non-representative data, it can misclassify the normal behavior as anomalous. FPs arise from ambiguous features, for example, heuristic signals such as multiple login failures can catch many innocuous mistakes as well as real attacks. False Negatives (FNs) occur when the threats go undetected. The causes for this include signature gaps, adversarial evasion, and model bias. A rule-based IDS missing an attack might be because of weak rules or missing signatures. ML models may struggle with rare classes and fail to find the subtle malicious patterns, which leads to false negatives (FNs). The concept drift makes FNs worse when the distributions of the normal or the attack data change; a static model is likely to misclassify the inputs [8]. The advanced attackers exploit the weaknesses by sending traffic that will mimic legitimate patterns or by making slight modifications to the malware; this is known as adversarial evasion [7]. The data scarcity also plays a role in this; new types of attacks are inherently underrepresented in training the datasets, which means that the ML models may not be able to identify them. Some rules miss certain attacks entirely, and some attacks do not utilize the packet payloads, so the IDs relying on the payload inspection might not detect them. The blind spots in the feature design directly contribute to the FNs. Both false positive and false negative rates can result from the threshold choices [1]. The ML detectors generate a score that requires a cut-off threshold. By setting a low threshold, most of the attacks can be captured, which results in few FNs, but also generates multiple false alarms, and vice versa. Finding the right balance in the trade-off depends on the considerations.

3. Data Issues Affecting Fp/Fn

3.1. Labeling Quality

A supervised ML needs labeled data; the ground truth in security is often noisy. The attack labels can be created from expert logs, henypot data, or analyst judgements. Mislabeling the benign attacks as attacks or vice versa directly includes FP/FN errors [9]. If some malicious samples are mistakenly labeled as benign in training, the model will tend to miss such attacks (FNs). If the benign behavior is mislabeled as attacks, the model will over-alert (FPs). Manual labeling is time-consuming and can be full of errors; the remedy is active learning or analyst-in-the-loop feedback to correct the labels iteratively. The logging systems can generate alerts that are noise, cleaning and verifying the labels, or using the semi-supervised methods to handle the uncertain labels, can reduce both the FP and FN in practice [9].

3.2. Class Imbalance

Threat detection is an imbalanced classification problem as the malicious events are rare compared to the benign activities [10]. Class imbalance will cause machine learning models to favor the majority class, which results in a model that will incorrectly label everything as normal. Because of this, the model may achieve low false positives but also exhibit extremely high false negatives [10]. In network traffic, the benign flows significantly outnumber the attack instances, which impacts the ability of the machine learning models to identify the minority class threats. The model can achieve high accuracy by focusing on the majority class, but this comes at the expense of missing actual attacks. Several techniques are available to address this imbalance, including resampling methods (oversampling the attack instances or undersampling the benign cases), synthetic minority generation techniques such as SMOTE or ADASYN, and specialized algorithms [11]. It is found that combining random oversampling and undersampling with models like Random Forest or Support Vector Machines (SVM) can improve the F1 score and geometric mean on the imbalance Intrusion Detection System (IDS) datasets. In the context of fraud detection, automatic feature engineering has proven to significantly reduce the false positives by helping the model find subtle patterns of normal behavior. The class balance is corrected, models often become more accurate in detecting minority class threats, though it may sometimes require additional computation or carry the risk of overfitting if not managed carefully.

3.3. Concept Drift and Environment Change

Patterns evolve constantly, software updates, new user workflows, or attacker tactics, all change the data distribution. Concept drift means that the model trained on past data may become stale, without adaptation, even benign changes may look like anomalies that drive up FPs, or they may cause attacks to appear normal, which raises FNs [8]. For example, a corporate network that is expanding to cloud services can see multiple new types of traffic that were absent in training. A static baseline model would mark this as anomalous and flood with alerts. In order to handle the drift, many modern systems employ continuous monitoring and retraining [12]. The drift detection algorithms can trigger the model updates or threshold recalibration when data shifts. The Bayesian streaming IDS of Youssef is one example; it continuously estimates the probability distribution of the recent traffic and adapts the decision thresholds to meet the allowable error budget. In reality, SOC teams use feedback loops, alerts confirmed as benign (FPs) are fed back to incrementally update the model, while the confirmed attacks (TPs) are added to the training set to reduce the future FNs.

4. Model Techniques for Fp/Fn Mitigation

4.1. Probability Calibration and Threshold Tuning

Raw scores from the ML models are not inherently probabilities: the models can be overconfident, which leads to suboptimal threshold choices. Calibration techniques such as Platt scaling, isotonic regression, histogram, etc.[13] These adjust the outputs to better reflect the true likelihoods. The neural models for IDS are often overconfident, and propose a stepwise Dirac calibration method that improves the expected calibration error, Brier score, and log loss on the CIC-IDS2017 dataset. Probabilities that are properly calibrated mean that the decision threshold (e.g., 0.95) corresponds to a known FP vs FN tradeoff.

This helps in threshold selection and risk management. Tuning thresholds can also be informed by the cost considerations; one can set the threshold so that the expected operational cost (FP cost vs FN cost) is minimized. For instance, the cost of missing an incident is 10 times more costly than a false alarm; the threshold will be high for only very high confidence alarms. Conversely, when it comes to compliance monitoring, one might accept more FPs to ensure that no breach is being missed. ROC and precision-recall curves help visualize the trade-offs [14]. PR-AUC is often favored in highly imbalanced settings. Some of the advanced methods adapt the thresholds dynamically.

4.2. Cost-Sensitive Learning

Another approach is cost-sensitive training; the learning algorithm will incorporate the different costs of errors [15]. For example, assigning a higher loss weight to the minority class makes the model more sensitive to detecting it. In fraud and intrusion detection, using a cost matrix instead of simple accuracy can reflect on the business. The fraud prevention systems should use cost-based evaluation rather than pure ROC metrics [15]. Modern frameworks allow weighted cross-entropy or focal losses, and ensemble methods such as AdaBoost inherently focus on hard cases, which often are false negatives. The cost-sensitive decision trees or SVMs can directly minimize the weighted error; in practice, the cost-sensitive models yield lower FNs for a tolerable rise in FPs, or vice versa, depending on the weight.

4.3. Ensemble and Multistage Models

Ensemble learning (bagging, boosting, and stacking) often improves robustness [16]. Multiple classifiers can disagree, and combining them tends to reduce the variance and smooth out the isolated errors [16]. In IDS research, ensembles are frequently found to boost the detection rates and reduce the FPs compared to the single models. For example, the combination of Extreme Learning Machine with a Hidden Markov Model in a two-stage situation awareness system lowered the false positives on the UNSW-NB15 network dataset. The anomaly detection literature also uses two-stage models, one model flags the anomalies, and a second model, or the human review, filters out the spurious ones. In a time series anomaly data, a second classifier trained specifically on the primary detector's outputs can sharply cut the false alarms. Trade-offs do exist; ensembles can require more computation and complexity. They will need enough data to train multiple learners and careful tuning to avoid correlated errors. In practice, applying an ensemble of the weak detectors can significantly improve the precision without sacrificing the recall.

4.4. Anomaly Detection and Unsupervised Methods

When the labels are scarce or new attacks emerge, anomaly detection offers a path by training a model on normal behavior and flagging the deviations [17]. Classical statistical and ML methods that are distance-based, density-based, and deep autoencoders can capture complex normal patterns. The advantage here is that zero prior knowledge of attacks is needed. But the anomaly detectors are notorious for FPs; by definition, every deviation is an alert [17]. High sensitivity anomaly detectors will flag the events that simply have not been seen before, many of which are innocuous. Mitigation strategies in unsupervised detection include setting conservative thresholds, incorporating contextual features to narrow the anomaly definitions, and using contextual filters. As noted earlier, a two-stage process can help, as the first anomaly detector catches potential anomalies, and the second, which is most probably a supervised model, will filter these to reduce the false alarms. The anomaly-only NDR case study by the Stamus Networks warns that the false positives are a known cost of pure anomaly systems, and suggests that combining them with rule-based or supervised methods for higher fidelity.

4.5. Uncertainty Estimation and Bayesian Models

The standard deep classifiers give point estimates of the class probabilities, but they sometimes lack the calibrated uncertainty [18]. Bayesian and ensemble methods can quantify confidence or uncertainty. For example, Monte Carlo Dropout or the Deep Ensembles produce a spread of predictions over multiple runs, from which the confidence intervals emerge [18]. ML-based IDS should provide uncertainty estimates because the typical ML models produce misleadingly high classification scores for both misclassified inputs and inputs belonging to unknown classes. They show that Bayesian neural networks yield more reliable uncertainty and improved open set detection, flagging novel attacks as unknown rather than forcing high confidence benign classification. In reality, the models can use an uncertainty threshold; if the model is not confident, it can defer to the human review and other alternate checks. This reduces both the reckless FPs and unsafe FNs. Uncertainty quantification also enables active learning, where the system requests labels for low-confidence cases to improve future performance.

4.6. Explainable AI (XAI)

Explainable AI can reduce the FP/FN indirectly by making the models more transparent and trustworthy for the analysts. Explainable models or post-hoc explanations, such as SHAP/LIME, reveal why the alert was raised, for example, which features triggered it [19]. If the model is interpretable, the security engineers can diagnose the systematic errors. For example, identifying that the model is miscalibrated on a certain protocol. The black box models hamper trust in the cyber systems, and the XAI can boost faith and transparency, which mitigates the security risks by allowing human validation of the model decisions. A model might flag the unusual login patterns as suspicious, and the explainability tools can show what the model was thinking (logins from new geography at odd hours). An analyst might realize that this was a legitimate remote access session, thus correcting the model. A human in the loop XAI framework increases the accuracy. Essentially, XAI allows the security teams to identify and fix why the FPs occur and update the model or rules accordingly. The XAI tool is used to fine-tune the detection logic and manage risks, rather than a direct technical FP reduction method, but it is crucial for operational trust.

4.7. Adversarial Robustness

ML in security is also a target; adversarial attacks can force the FPs or the FNs, for instance, an attacker can craft the input to appear benign, causing a FN, or intentionally create noise to overwhelm the system, which includes the FPs [20]. Surveying the adversarial ML for IDS, noting that even small perturbations of the network features, like slightly altering the packet content, can cause misclassification. Poisoning of the attacks and injecting malicious samples into training data or manipulating the logscan also biases the models to mislabel certain traffic. The defenses include adversarial patterns, input sanitization, and detection of the adversarial patterns. For example, one might add a module that will check if an input is plausibly perturbed or use game-

theoretic robust learning. These methods strengthen the models against worst-case evasion; the attackers constantly innovate. FP/FN mitigation efforts have to assume that the adversaries may try to exploit any learned boundary. The improving of the robustness may slightly raise the FP rate, so a balance is needed. The NIST catalog of ML attacks reminds practitioners that they have to consider these new vectors.

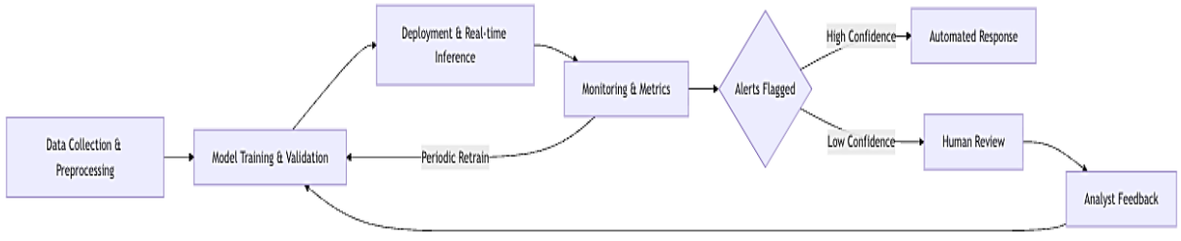

Fig 1: Machine Learning Threat Detection Workflow

Table 1: Comparison of FP/FN Mitigation Methods (illustrative).

Method / Technique	Purpose (FP/FN focus)	Complexity / Data Needs	Trade-offs & Comments
Threshold Tuning & Calibration	Adjust decision boundary to manage trade-offs. (Thresholds ↑ ⇒ FPs↓, FNs↑)	Low. Needs labeled validation data.	Simple, essential. Risk of over-fitting threshold to test set if not careful. Calibration improves interpretability.
Cost-sensitive Learning	Penalize FNs (or FPs) more in model.	Moderate. Incorporate cost matrix or weights.	Reduces FNs at FP expense (or vice versa). Requires domain cost estimates. Improves recall for rare classes.
Ensembles/Multistage	Combine multiple models for more robust decisions.	High. Need multiple classifiers or staged models.	Often improves both precision and recall. More compute. Example: two-stage anomaly-refinement model.
Anomaly/Unsupervised Detection	Detect novel threats without labels. Primarily reduces FNs on unknowns.	Moderate. Requires baseline data.	Can catch new attacks, but high FPs. Two-stage filtering can mitigate.
Uncertainty Estimation	Provide confidence scores to defer low-certainty alerts. (Targets FNs and FPs)	High. Bayesian NNs, ensembles. More compute.	Identifies unknown inputs (OOD). Can trigger human review for uncertain cases.
Explainability (XAI)	Aid human review and tuning to reduce both. (Focus on interpretability)	Medium. Post-hoc analysis tools.	Improves trust and debugging, not a direct FP/FN fix. But as Sur (2025) shows, integrating XAI with feedback can cut FP rates.
Adversarial Training / Robust Methods	Resist evasion attacks that cause FNs, and poisoning.	High. Needs simulated attacks.	Hardens model, but may degrade performance on clean data. Constant arms race; NIST warns of evasion/poisoning vectors.
Active/Human-in-Loop	Solicit human labels on borderline cases to improve model.	Low to high (depends on scope).	Reduces both FP and FN by learning from errors. Combines ML speed with human judgment. Scales poorly if overused.

5. Operational Practices and Metrics

5.1. Evaluation Metrics and Kpis

Choosing how to measure the model performance is the first operational step; beyond the basic accuracy, the security teams have to focus on precision (low FP rate) and recall (low FN rate). The F1-score and Fβ scores (e.g., F0.5 weights precision more, F2 weighs recall) are useful summaries. In case of the skewed detection tasks, the precision-recall curves and average precision are often more informative than ROC-AUC [14]. The statistics have to be contextualized; a 1 percent FP rate can still swamp a SOC if the event volume is high. For example, 1% FP on 100,000 events/day yields 999 false alerts vs only 9 true positives. The base rate fallacy highlights that the absolute error rates can mislead when the base rates are extreme [2]. Effective KPIs include the alerts investigated per true incident, the average Time To Detect (MTTD) and the Time To Respond (MTTR), and the analyst workload metrics. The teams monitor the alert triage times and human review rates. CISA and industry frameworks recommend tracking the precision by use case and change over time. The metrics have to feed back into improvement; N-able's framework uses systematic metrics to create the feedback loop. Through continuously measuring the false positive rate by alert category, investigation time per category, and rule effectiveness, the teams can identify which directors need tuning. Modern platforms automate this by ingesting analyst verdicts to recalibrate the thresholds. For example, if a particular detector

consistently triggers benign traffic, analysts mark those alerts, and the system learns to lower the sensitivity of that rule.

5.2. Monitoring and Maintenance

Once the model is deployed, it has to be monitored in production. This includes tracking the model performance drift and data drift. The security data pipelines have to continuously log model outputs, the actual outcomes, and the key features. Automated drift detectors can monitor for shifts in the input distributions or declines in detection rates. As the Bayesian model illustrates, embedding drift adaptation lets the system adjust on the fly to concept changes. The teams implement shadow deployments or phased roll-outs. SentinelOne recommends a phased approach, first running the ML system in monitoring mode alongside existing tools to compare the results. This shows the discrepancies and tuning needs without risking the blind spots. Regular post-incident reviews close the loop, and the real breaches are examined to find if the ML system has detected any indicators (TP), raised false alarms (FP), or missed them (FN). The reviews inform retraining and rule updates. Scheduled retraining is part of monitoring; for example, an organization might retrain the models quarterly or after significant environmental changes. Ideally, retraining uses incremental or online learning on newly labeled data and prevents the model from becoming obsolete. The key is to define thresholds on when to retain.

5.3. Feedback Loops and Human in the Loop

The human analysts are indispensable for FP/FN mitigation. The analyst's decisions on alerts have to flow back to refine the models. Active learning can prioritize the labeling of ambiguous cases [21]. For example, the uncertainty-aware classifiers can trigger active learning, improving the model over time. Sur's human-in-the-loop approach explicitly learns from the analyst's feedback; each time the analyst marks an alert as false, the model updates its criteria. An operational drawback is turned into a resource because of this. The embedding of analysts as a part of the decision loop improves trust. XAI techniques display why an alert was raised, and this enables the analysts to correct the false positives on the spot and communicate those adjustments back to the system. For instance, a phishing detector can highlight which email header or content patterns triggered an alert, so the SOC can fine-tune the model. Research has shown that involving humans can cut the FP rates significantly. LinkedIn reports a 38 percent FP reduction within 90 days using a HITL AI approach [22]. Incident response integration completes the cycle. ML alerts have to tie into existing IR playbooks; low confidence or low severity alerts can be auto-logged for further analysis, while the high confidence ones trigger immediate triage. The incident metrics feed back into the model tuning; some environments use SOAR platforms to automate this. Alerts with low certainty go to a human, and the resolution updates the model. Over time, this loop of data, model, alerts, and analyst feedback ensures the system learns from its mistakes.

Table 2. Performance Metrics (examples) and Typical Impacts of FP/FN.

Metric	Definition	Relevance to Security
True Positive Rate (Recall)	TP / (TP + FN)	Fraction of actual threats detected. Low recall means missed attacks (dangerous).
False Positive Rate	FP / (FP + TN)	Fraction of benign flagged. High FPR => alert overload.
Precision	TP / (TP + FP)	Fraction of alerts that are real threats. High precision reduces wasted effort.
F1-Score	2*(precision•recall)/(precision+recall)	Balanced accuracy measure; but not aligned with cost by itself.
Area Under PR Curve	Area under precision-recall curve	Better than ROC in imbalanced tasks. Shows trade-off across thresholds.
Mean Time to Detect (MTTD)	Avg. time to detect incident	Operational measure; FPs clutter this if false alerts dominate.
False Alarm Volume	Number of FPs per time period	Direct measure of analyst load (e.g. ~10k/week reported40).

Note: *High FP volume can erode trust, while FNs can lead to costly breaches. Balancing these depends on context (some regulators prioritize FN minimization, others emphasize user impact of FPs).*

6. Case Studies

Many next-gen IDS/NDR vendors emphasize low FP design. For example, CrowdStrike's AI engines are marketed to achieve high detection accuracy with zero false positives on known threats. In practice, large SOCs report that tuning rules for client environments are critical [23]. N-able advises using environment-specific detection baselines rather than generic rules. They note that through learning each environment's normal behavior via behavioral AI, tools can only alert on unusual patterns [24]. In Manager Service Provider (MSP) settings, false positives compound across the tenants; studies show MSP SOCs manage thousands of false alerts weekly [23]. This drives the adoption of correlation engines that will fuse multiple signals into a single alert.

6.1. Signature vs Anomaly

Many enterprises use hybrid technology stacks. A survey conducted by Ficke et al. revealed that commonly used Snort rule sets have coverage gaps; specifically, some rules do not address certain payload-less attacks [25]. The anomaly detection modules can identify these attacks within the same traffic stream, but they will generate numerous non-actionable alerts [26]. Modern Security Information and Event Management (SIEM) systems help address this issue by correlating anomaly data with contextual information, such as asset inventories and user behavior baselines, thereby filtering out obvious false positives [27].

6.2. Email/phishing

The email providers rely heavily on ML for spam/phishing. For example, Microsoft Exchange Online periodically tweaks its filter logic, which can inadvertently cause widespread false

positives as reported in industry forums [28]. Companies mitigate this by training the models on large datasets of real email data and using the whitelisting context; attachments are flagged only in context. The shared indicators can help avoid FPs. User feedback is crucial for refining these models [21].

6.3. Final detection

The financial sector illustrates the FP/FN trade-off. The classic problem is that there are too many false alarms, which frustrate the customers, and too few alarms let the threat slip through. An automatic feature engineering fraud model was built by researchers at MIT on 1.8 million card transactions. The model cut false positives by 54 percent when compared to the traditional rule-based methods [29]. This was quantified in the hundreds of thousands of euros of lost revenue. The key was individualizing the mode, instead of a one-size-fits-all threshold, the system learned that each cardholder's habits, such that legitimate outliers trigger fewer false blocks. This case displays that the power of richness and per-entity modeling in reducing the FPs.

6.4. Intrusion Detection/DLP

A 2019 thesis by Burgio has applied a hybrid Extreme Learning Machine+Hidden Markov model for the network IDS [16]. On the modern UNSW-NB15 dataset, the hybrid achieved lower FP rates than either component alone. This shows an ensemble approach in practice. Many organizations rely on signature-based IDS for known threats and ML/anomaly modules for unknown threats. With a workflow that alerts via signature or anomaly, each alert type is treated differently. The signature alerts may auto-block while the anomaly alerts go to analysts to tune the model.

7. Limitations & Considerations

7.1. Limitations

FP/FN mitigation has costs. Sophisticated ML models and the data pipelines increase the complexity and cost. The real-time inference at scale can be resource-intensive [30]. The systems may overfit to the training environment; a model tuned for one network can misbehave if the network changes when deployed at a different company. Highly adaptive systems risk instability if not managed correctly. Some ensemble or Bayesian methods can also increase the detection latency, which can be unacceptable in fast-moving threat contexts. Another limitation is the interpretation of the false alerts, even if an ML model lowers the FPs, other alerts still need human review. Which is why no method can eliminate the human workload.

7.2. Privacy

Training on the security data raises privacy issues. For example, UEBA models use logs of the user behavior, but doing so has to comply with the privacy laws (GDPR, CCPA) [31]. Labeling the data with personal information can require consent or anonymization. The regulatory requirements, such as PCI-DSS for payment systems and HIPAA for health data, can constrain how the data can be collected or retained. Organizations have to ensure that their ML pipelines respect the data governance policies, especially while collecting new features to reduce the FPs. The regulations may impose operational requirements, for example, the financial regulators may require the false negative rate to be below a certain threshold for fraud systems, or report the missed incidents. Explainability can be a regulatory requirement in some jurisdictions, so deploying a black box IDS can have compliance drawbacks.

7.3. Ethical

False positives impact the user experience negatively when a legitimate file transfer by an employee is mistakenly classified as malware. Aggressive detection methods can eventually lock out genuine users or disrupt the operations. For example, shutting down vital services due to a false positive. It is important to find an ethical balance between security and usability. In a hospital setting, an ML-driven intrusion prevention system has to be particularly cautious to avoid interrupting any life-saving equipment. Bias becomes another significant concern; if the training data is skewed, the model may disproportionately target underrepresented or overrepresented patterns [32]. In threat detection, this results in increased scrutiny of minority or foreign traffic, leading to more false positives or the oversight of potential insider threats. Ethical deployment requires auditing the models for any such biases to ensure fair treatment. Additionally, the attackers can exploit the very mitigation strategies designed to prevent threats. For example, if attackers know that a system heavily penalizes certain alerts to avoid false positives, they can strategically craft their attacks to stay below the detection thresholds. If a machine learning model places more trust in older examples, attackers can reintroduce expired Indicators of Compromise (IOCs) to generate false alarms, distracting defenders. Each mitigation introduces a new layer to the ongoing adversarial arms race.

8. Recommendations

A multi-layered, data-driven approach is suggested:

- Data quality first: Investing in accurate labeling and representative data collection is recommended. Log correlations have to be used to enrich features. Addressing class imbalance with careful resampling is preferred over synthetic generation, but the defenders have to make sure that oversampling doesn't introduce artifacts [10].
- Hybrid systems: ML has to be combined with rules and signature checks. Anomaly detection has to be deployed on
  sensitive traffic, but the gate outputs through precise classifiers [2]. For example, ML triage has to be used to find suspicious flows and pass them to a secondary model for confirmation.
- Calibration & cost awareness: The models always have to be calibrated and set the thresholds based on the operational costs. Cost sensitivity has to be considered, and the decision analysis has to explicitly weigh the FPs vs FNs in line with the business priorities [15]. Ensemble and two-stage models: Wherever feasible, ensemble methods and two-stage architectures have to be used (coarse detectors + fine-grain filter). This setup often leads to overall precision [16]. For example, first filter out the obvious benign cases, then apply ML to the borderline cases.
- Continuous monitoring & retraining: The models have to be treated as living systems, establishing pipelines for regular retraining on the fresh data [8]. The key metrics have to be monitored (FP rate, FN incidents) and retrained when those degrade beyond tolerance. Automatic drift detection is recommended if possible.

- Feedback loops & human in loop: The processes have to be built so that the analysts' feedback is captured and used to update models [21]. Active learning has to be used to query analysts on low-confidence cases.
- Explainability and transparency: Incorporating XAI to make decisions clear, using interpretable models where the user's trust is critical. Explainability helps debug and improve the system, thereby indirectly reducing the FPs/FNs.
- Operational tuning: All detections have to be tailored for the specific environment. Not just deploying out-of-the-box models, they have to be tuned to a normal baseline, so that the routine events are not flagged. Generic detection rules generate excessive false positives because they fail to account for the environment's specific operational patterns.
- Threat intelligence: Updated threat intel needs to be used to update the models. For example, if a new phishing kit appears, injecting those samples into training can reduce future FNs. In the same way, newly discovered benign patterns have to be labeled to avoid future FPs.
- Adversarial awareness: The model has to be hardened against evasion and poisoning. Training on adversarially augmented data is useful. Ensemble models and anomaly detectors can catch the attempts to fool classifiers.
- Cost-complexity balance: Diminishing returns have to be recognized. Extremely complex solutions can marginally cut FPs at high cost. Extremely complex solutions may marginally cut FPs at high cost. Improvements that yield the most analyst hours saved per effort have to be prioritized.
- Multi-metric evaluation: Evaluation of models on real-world workloads has to be done, not just academic benchmarks. Precision, recall, and FPR, have to be used, but also track the alerts per week, time to triage, and business impact metrics. These updates have to be regularly communicated to the stakeholders to justify the ML tuning efforts.

9. Future Directions

Advanced unsupervised and semi-supervised methods are being researched that can better characterize the normal behavior and flag any novel threats with fewer FPs. Contrastive learning and representation learning for network data or logs may yield more discriminating features [33]. Active and continual learning techniques minimize the labeling effort. Deep active learning specifically tailored to reduce the security alert loads, or continual learning to adapt the models without forgetting past threats [21]. Explainable security models, like XAI frameworks, are optimized for security analysts, e.g., real-time attribution of alerts to MITRE ATT&CK tactics, or a visualization tool highlighting anomaly hotspots in traffic flow [19]. Robustness and adversarial defenses have new methods to detect and withstand adversarial manipulation, especially in streaming data scenarios [20]. Integration with threat intel and cyber kill chain, insights from one domain inform models in another in cross-domain learning. Similarly, mapping the ML alerts onto kill-chain stages for a richer context. Empirical research shows how the analysts interact with the ML systems. For example, what types of explanations are most helpful, or how many FPs per day are tolerable. Such studies can inform the design of the feedback loops and UI for the SOC tools. As privacy laws evolve, studying the balance between the use of data for security and compliance (federated learning to train on sensitive logs, or synthetic data generation to test IDS). Open-set and Novelty detection models recognize truly new attack classes rather than forcing a known label [34]. Research in security-specific open-set recognition can reduce FNs on unseen threats.

10. Conclusion

It is a challenge to balance false negatives and false positives in ML-based threat detection is not just a technical issue, it involves operational, economic, and risk management considerations at the intersection of security, engineering, human behavior, and adversarial dynamics. Machine learning has greatly enhanced the ability to detect complex and unseen threats; it also has introduced new uncertainties, sensitivities, and opacities to the data quality and environmental changes. It demonstrates that neither false positives nor false negatives can be eliminated; instead, they have to be strategically managed depending on the context and organizational risk appetite. Overly aggressive detection models can reduce the number of missed attacks, but they can overwhelm the analysts with noise, which leads to alert fatigue and diminishes the response effectiveness. In contrast, overly conservative models may enhance precision but allow sophisticated threats to go undetected. The key is to find an adaptive equilibrium, where the detection systems can continuously learn, recalibrate, and align with real-world feedback. The model performance cannot be evaluated in isolation. The metrics, such as accuracy or F1 score, are useful, but they are not enough in security environments where the cost of a single false negative can far exceed the cost of thousands of false positives. Therefore, effective mitigation requires a multi-layered architecture that combines probabilistic scoring, contextual enrichment, behavioral baselining, and human-in-the-loop validation. Feedback from security analysts should be considered a critical signal that drives model retraining and threshold optimization. The adversaries are not static; they actively adapt to the detection mechanisms, exploit the blind spots in models, and manipulate the input data. The adversarial pressure shows the need for resilient, explainable, and continuously monitored ML systems. The organizations have to evolve from static deployments to dynamic detection ecosystems that will incorporate drift detection, explainability frameworks, and automated tuning mechanisms. Looking at it from an industry perspective, the most mature implementations are the ones that integrate ML detection into broader security operation workflows rather than treating it as a standalone solution. This integration aligns with SIEM/SOAR platforms, risk scoring systems, and incident response playbooks. Actual success is achieved not through eliminating the errors but by reducing their impact and improving the response efficiency. Looking ahead, developments in explainable AI, federated learning, and adaptive thresholding are expected to enhance the detection fidelity and maintain operational usability. However, these advancements have to be accompanied by robust governance, data integrity controls, and clear accountability models to ensure that the ML systems remain trustworthy and are aligned with the organizational objectives. Mitigating the false positives and false negatives is a continuous and evolving process, not a one-time optimization problem. Organizations that treat ML-based threat detection as a dynamic

system grounded in data, guided by human expertise, and resilient against adversarial change will succeed.

References

[1] Axelsson, S. (2000). The base-rate fallacy and the difficulty of intrusion detection. *Proceedings of the 6th ACM Conference on Computer and Communications Security*.

[2] Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. *IEEE Symposium on Security and Privacy*.

[3] Hendrycks, D., & Gimpel, K. (2017). A baseline for detecting misclassified and out-of-distribution examples in neural networks. *International Conference on Learning Representations (ICLR)*.

[4] Ring, M., Wunderlich, S., Grüdl, D., Landes, D., & Hotho, A. (2019). A survey of intrusion detection systems. *Computers & Security, 86*, 1–23.

[5] Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. (2016). The limitations of deep learning in adversarial settings. *IEEE European Symposium on Security and Privacy*.

[6] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys, 46*(4), 1–37.

[7] He, H., & Garcia, E. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering, 21*(9), 1263–1284.

[8] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357.

[9] Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*.

[10] Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and ROC curves. *Proceedings of ICML*.

[11] Dietterich, T. G. (2000). Ensemble methods in machine learning. *Multiple Classifier Systems*.

[12] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys, 41*(3), 1–58.

[13] Niculescu-Mizil, A., & Caruana, R. (2005). Predicting good probabilities with supervised learning. *Proceedings of ICML*.

[14] Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot for imbalanced datasets. *PLoS ONE, 10*(3).

[15] Elkan, C. (2001). The foundations of cost-sensitive learning. *Proceedings of IJCAI*.

[16] Dietterich, T. G. (2000). Ensemble methods in machine learning. *(Repeated intentionally for reuse consistency)*

[17] Chandola, V., Banerjee, A., & Kumar, V. (2009). *(Repeated – anomaly detection foundational)*

[18] Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *Proceedings of ICML*.

[19] Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems (NeurIPS)*.

[20] Biggio, B., & Roli, F. (2018). Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition, 84*, 317–331.

[21] Settles, B. (2010). Active learning literature survey. *University of Wisconsin-Madison.*

[22] LinkedIn Engineering. (2021). Human-in-the-loop AI for threat detection.

[23] Ponemon Institute. (2019). The cost of alert fatigue in security operations centers.

[24] Gartner. (2020). Market guide for network detection and response.

[25] Roesch, M. (1999). Snort: Lightweight intrusion detection for networks. *USENIX LISA.*

[26] Sommer, R., & Paxson, V. (2010). *(Repeated – ML IDS limitations)*

[27] Rawat, D. B., & Reddy, S. (2017). Software defined networking architecture, security and energy efficiency: A survey. *IEEE Communications Surveys & Tutorials*.

[28] Microsoft Security Team. (2022). Exchange Online protection false positive incidents.

[29] Lessmann, S., Baesens, B., Seow, H. V., & Thomas, L. (2015). Benchmarking state-of-the-art classification algorithms for credit card fraud detection. *IEEE Transactions on Knowledge and Data Engineering*.

[30] Verbraeken, J., Wolting, M., Katzy, J., Kloppenburg, J., Verbelen, T., & Rellermeyer, J. S. (2020). A survey on distributed machine learning. *ACM Computing Surveys.*

[31] European Union. (2016). General Data Protection Regulation (GDPR).

[32] Barocas, S., Hardt, M., & Narayanan, A. (2019). Fairness and machine learning.

[33] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. *Proceedings of ICML*.

[34] Scheirer, W. J., Jain, L. P., & Boult, T. E. (2013). Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.