Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

AI-Driven Threat Intelligence: Evaluating Machine Learning for Real-Time Cyber Threat Sharing Among U.S. National Security Agencies

Mohammed Nazmul Islam Miah[1]✉, Md Joshim Uddin[2], and Md Wasim Ahmed[3]
[1]*Master of Public Administration, Gannon University, Erie, PA, USA*
[2]*Master of Law, ASA University of Bangladesh*
[3]*Master of Law, Green University of Bangladesh*
Corresponding author: Mohammed Nazmul Islam Miah. **Email:** islammia001@gannon.edu

| ABSTRACT
This study explores how artificial intelligence, specifically machine learning and federated learning, can support secure and real-time threat intelligence sharing among national security agencies in the United States. The core idea was to evaluate whether decentralized machine learning systems could help multiple agencies detect and respond to cyber threats more quickly, without forcing them to share sensitive raw data. The approach was built in three phases. First, we trained several supervised learning models independently on each agency's data to understand their predictive capabilities. That gave us a baseline for how each agency's threat signals behaved in isolation. In the second phase, we introduced a federated learning setup, allowing models to be trained collaboratively across agencies without data ever leaving its original environment. This was combined with privacy-preserving techniques like secure aggregation and differential privacy to meet the high-stakes security demands of national defense. The third phase focused on explainability, using SHAP values to interpret model predictions and help agencies understand not just what the model predicted, but why. What stood out was that while individual models showed promising results, their performance and generalization improved substantially in the federated setup. And when explainability was layered in, the models became more trustworthy, helping bridge the gap between AI automation and operational decision-making. This isn't about just building smarter threat models. It's about enabling a shift from siloed, reactive defense to a more coordinated, real-time security posture. The architecture we tested is not purely theoretical; it's a practical framework that could be deployed in government settings today. As cyber threats grow in complexity and speed, so must our tools for responding to them. This study shows that AI can be part of that shift, not by replacing human analysts, but by giving them faster, clearer, and more secure ways to see what's coming next.

| KEYWORDS
Threat Intelligence, Federated Learning, XGBoost, MLP, SHAP, Secure Aggregation, Differential Privacy, National Security, Explainable AI, Cybersecurity Collaboration

| ARTICLE INFORMATION
ACCEPTED: 12 July 2025 **PUBLISHED:** 03 August 2025 **DOI:** 10.32996/jcsts.2025.7.8.34

1. Introduction

1.1 Background and Motivation

You've probably noticed how cyber threats against the U.S. national infrastructure aren't isolated blips anymore. They've morphed into elaborate, multi-pronged operations that can knock out entire networks, sneak into sensitive government systems, and even shake public trust. In that setting, federal agencies detecting and sharing threat intel quickly and securely matter more than ever. Yet, despite everyone knowing it's critical, old-school systems, overlapping authorities, strict classification rules, and data sensitivity keep agencies locked in silos. The result? Missed early warnings, effort duplication, and slow, disjointed responses. Enter machine learning, and more specifically, federated learning. Instead of gathering all data in one place, federated learning lets different groups train a shared model on their own turf. Sensitive data stays put, legal and operational limits remain intact, and everyone

still benefits from collective improvements. We've seen in finance how tree-based models unearth hidden patterns in transactional data for fraud spotting (Jakir et al., 2023)[12], and how hybrid frameworks boost threat classification in financial security (Sizan et al., 2025)[20]. There's work showing pattern-detection methods catching complex scams in crypto transactions (Das et al., 2025)[5].

On top of that, Islam and colleagues (2025)[11] demonstrated cleaning up unstructured, noisy e-commerce logs, proving that these approaches can handle the messy, high-volume nature of cybersecurity data. Rahman et al. (2025)[18] make a case for using distributed ledgers like blockchain to add transparency across supply chains, a concept that maps neatly onto the inter-agency sharing problem. And Sultana et al. (2025)[22] sketch out how decentralized architectures plus machine learning can power efficient edge computing, hinting at the infrastructure model we'll need for secure, real-time collaboration. Taken together, these studies point toward a scenario where federated learning, backed by rigorous privacy and security measures, could let agencies detect threats together in real time while preserving each one's autonomy. It's not a pipe dream: it's the logical next step for a landscape where speed, coordination, and data protection all have to play nice.

1.2 Importance of This Research

Securing the digital backbone of national security agencies goes beyond ticking technical boxes; it's about keeping operations running smoothly, even when threats evolve at breakneck speed. Cyber adversaries are no longer lone hackers; they act in concert and shift tactics on a whim. That means piecemeal defenses fall short and agencies need to pool their insights, adapt on the fly, and coordinate responses without exposing their most sensitive data. This research cuts to the heart of that puzzle: how to collaborate in real time without trading away privacy or independence. Machine learning brings something special to the table here. It can uncover hidden patterns, learn from messy data, and keep pace with emerging threats in a way that rigid, rule-based systems can't. Even better, explainable AI shines a light on why a model flags something as suspicious, an absolute must when lives or national interests are on the line. You get predictions alongside clear reasoning, which helps build the trust and accountability agencies need.

Federated learning tackles the biggest roadblock to sharing intelligence: no agency has to give up its raw data. Each one trains models on its own incident logs and security events, then merges insights into a collective model. This preserves each agency's boundaries while delivering a bird's-eye view of the threat landscape. The result is fewer duplicated efforts, faster, more coordinated responses, and a framework that grows and adjusts with new data or changing policies. In an environment where threats cross borders in seconds, being able to act immediately, without compromising data integrity, is vital. This study lays out a clear, technically sound path to make that happen.

1.3 Research Objectives and Contributions

This study looks into whether machine learning, specifically federated learning, can offer a secure and workable way for U.S. national security agencies to share threat intelligence without giving up control of their data. The big question driving the work is whether a decentralized model training can deliver strong, generalizable results without requiring all the data to be pulled into a central location. That matters a lot in settings where confidentiality, legal restrictions, and operational boundaries make direct data sharing nearly impossible. To test this, the study builds predictive models using each agency's dataset, then runs those models through a federated setup to see how well they hold up across different data environments. The idea is to assess whether these models can learn something useful from each other, even while the data stays put. To keep everything secure, the setup includes safeguards like differential privacy and secure aggregation. These are meant to protect each agency's input from being reverse-engineered or exposed, while still allowing the broader model to benefit from what each agency knows. SHAP is used after the fact to help explain why the models make certain predictions. That level of interpretability is important, especially if different agencies are going to trust the system enough to rely on it during active threat investigations or response efforts.

The study makes a couple of contributions. On one hand, it proposes a way to collaborate across agencies without crossing any privacy or policy lines. On the other hand, it delivers a working pipeline, a combination of ensemble learning, secure model protocols, and explainable AI that others could use. It also benchmarks how models like XGBoost and multilayer perceptrons perform in both federated and non-federated setups, giving some real-world guidance for anyone looking to build something similar. Finally, by using SHAP to tie together local insights into a broader national threat picture, the work suggests a way to turn fragmented intelligence into something more unified. The whole system is designed with scale, transparency, and interoperability in mind, so it could evolve alongside future security needs.

2. Literature Review

2.1 Traditional Cyber Threat Detection Approaches

More recently, machine learning has stepped in to fill the gaps. Algorithms like Random Forests, XGBoost, support vector machines, and deep neural nets have proven effective at teasing out patterns in messy network data, even when malicious traffic is a tiny fraction of the total. Ensemble methods give you resilience against noisy inputs and let you see which features matter most, which

is handy when you're investigating an alert. SVMs shine when training examples are scarce, since they focus on boundary detection. Neural networks, whether simple MLPs or more advanced recurrent architectures, can learn complex and time-dependent behaviors in flow logs. When you train these models on attack traces or telemetry from real corporate networks, they tend to catch intrusions, malware outbreaks, or lateral moves faster than fixed rules ever could. Tree-based learners love the structured features we extract, packet counts, flow durations, and byte ratios, while sequence models pick up on subtle shifts in traffic over time. That said, most deployments assume you can haul all your data into one spot to train a central model. In federal or military contexts, where data is locked behind strict access controls, that simply isn't possible. Even though ML delivers higher recall and adapts to new threats, its gains are curbed when datasets remain siloed behind agency firewalls.

As threat landscapes evolved, researchers started turning to semi-supervised and unsupervised techniques that don't rely on labeled data. Sommer and Paxson showed how clustering features from network traffic could reveal groups of suspicious activity even without knowing what the attack looked like beforehand (Sommer et al., 2010) [16]. That idea opened the door for models like autoencoders, which learn a compressed version of normal traffic and flag anything they can't reconstruct well. It's especially useful in settings where labeled examples are scarce or attackers constantly change tactics. More recent work adds techniques like one-class SVMs and contrastive learning, both of which can adapt as new patterns of legitimate behavior appear. Transfer learning also plays a role here: you can pre-train models on large public datasets like CICIDS2017, then fine-tune them with a smaller set of agency-specific logs. That cuts down on training time and lowers the data burden. Because many of these models work with abstract feature representations instead of full traffic dumps, they also fit well into privacy-sensitive environments. It means agencies can still benefit from shared knowledge without exposing sensitive internal data.

2.2 Machine Learning for Threat Detection

More recently, machine learning has stepped in to fill the gaps. Algorithms like Random Forests, XGBoost, support vector machines, and deep neural nets have proven effective at teasing out patterns in messy network data, even when malicious traffic is a tiny fraction of the total. Ensemble methods give you resilience against noisy inputs and let you see which features matter most, which is handy when you're investigating an alert. SVMs shine when training examples are scarce, since they focus on boundary detection. Neural networks,whether simple MLPs or more advanced recurrent architectures, can learn complex and time-dependent behaviors in flow logs. When you train these models on attack traces or telemetry from real corporate networks, they tend to catch intrusions, malware outbreaks, or lateral moves faster than fixed rules ever could. Tree-based learners love the structured features we extract, packet counts, flow durations, byte ratios, while sequence models pick up on subtle shifts in traffic over time. That said, most deployments assume you can haul all your data into one spot to train a central model. In federal or military contexts, where data is locked behind strict access controls, that simply isn't possible. Even though ML delivers higher recall and adapts to new threats, its gains are curbed when datasets remain siloed behind agency firewalls.

As threat landscapes evolved, researchers started turning to semi-supervised and unsupervised techniques that don't rely on labeled data. Sommer and Paxson showed how clustering features from network traffic could reveal groups of suspicious activity even without knowing what the attack looked like beforehand (Sommer et al., 2010) [16]. That idea opened the door for models like autoencoders, which learn a compressed version of normal traffic and flag anything they can't reconstruct well. It's especially useful in settings where labeled examples are scarce or attackers constantly change tactics. More recent work adds techniques like one-class SVMs and contrastive learning, both of which can adapt as new patterns of legitimate behavior appear. Transfer learning also plays a role here: you can pre-train models on large public datasets like CICIDS2017, then fine-tune them with a smaller set of agency-specific logs. That cuts down on training time and lowers the data burden. Because many of these models work with abstract feature representations instead of full traffic dumps, they also fit well into privacy-sensitive environments. It means agencies can still benefit from shared knowledge without exposing sensitive internal data.

2.3 Federated Learning in Cybersecurity

Federated learning adds another layer of complexity in cybersecurity, not only because of privacy, but also due to how different each agency's systems are. Some have stronger computer setups, others don't. Network conditions vary, data volume isn't uniform, and these differences can slow everything down, especially when lagging clients delay the overall update cycle. Kairouz and colleagues suggest a few ways to deal with this, including sampling clients adaptively and tailoring parts of the model to each agency's unique data [3]. They also point to communication strategies like gradient quantization and sparsification, which cut down the size of model updates by focusing only on the most important pieces or shrinking the amount of data sent. With the right error-feedback loops in place, these methods still allow the model to converge while easing the load on secure but bandwidth-limited channels. Adding in personalization layers helps each agency fine-tune the shared model to match its local threat profile. The result is a more resilient setup that balances the need for collective insight with the reality of different hardware, networks, and data distributions.

2.4 Explainable AI for Security Operations

As machine learning finds its way deeper into high-stakes security operations, explainability stops being a nice-to-have and becomes something you can't ignore. Security analysts, compliance officers, and policymakers aren't just looking for accurate predictions; they need to know why a model made a specific call. It's not enough for an alert to flag something as suspicious. People on the ground need to trust it, act on it, and later, be able to justify those actions if needed. This is where explainable AI tools like SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations) come into play. They help pull back the curtain, showing how different input features shaped a model's decision. In national security settings, where small anomalies can have outsized consequences, being able to trace back a model's reasoning is critical. A false alarm that no one can explain might slow everything down. A missed threat with no traceable logic? That's an accountability nightmare.

SHAP has gained a lot of traction because it's built on solid math and tends to be consistent. It doesn't just say which features mattered; it shows how much they mattered and how their influence changes across different predictions. That's key when you're deploying models in varied environments. Something like a spike in packet size might mean one thing at one agency and something completely different at another, depending on context like time of day or baseline behavior. SHAP's visuals, force plots, summary plots, and dependence plots help analysts tie model outputs back to what they know from the field. LIME, while narrower in scope, adds value too by testing how small changes to the input affect the output. It helps spot overfitting or weird correlations that the model shouldn't be picking up. That said, explainability in distributed or federated learning setups is still an open problem. Federated systems are messy by nature; different clients have different data, different model behaviors, and different definitions of what's normal. That makes it hard to generalize explanations across the board. SHAP values from one agency might not make sense somewhere else. Trying to merge those explanations into a unified global view gets complicated fast, both technically and operationally. And there's a privacy angle here too: in some cases, explanations themselves can leak sensitive patterns, especially if they hint at rare or unusual behaviors in the data. That risk goes up when the clients are government or defense entities, where secrecy is part of the job.

There's also a human side that's often overlooked. Even the most accurate explanation isn't helpful if the person reading it can't make sense of it. Security analysts are usually under pressure, and they're not going to stare at dense SHAP charts unless those insights are tied into their day-to-day tools. So, integration matters. Whether it's dashboards, automated alerts, or investigative workflows, explainability has to fit into how people work. There have been some interesting proposals, pairing SHAP with ontologies or incident taxonomies to make things more intuitive, but those ideas are still early, and there's no standard yet. And then there's the legal side. In national security, explanations aren't only for internal use. They may end up in audit trails, official reports, or even court cases. A flagged threat without a clear rationale probably won't hold up if you're trying to prove negligence or attribution. So explainability becomes more than a technical requirement; it's part of the broader structure of trust, accountability, and coordination across agencies. If federated learning is going to be viable in these environments, it's going to have to explain itself, not only accurately, but in ways that work across legal, operational, and policy boundaries.

2.5 Gaps and Challenges

Despite all the academic excitement around ML and federated learning, you rarely find end-to-end systems in production. Key hurdles remain: handling client drop-outs during secure aggregation, balancing noise for privacy against detection accuracy, and merging explanations from clients with wildly different data distributions. Most frameworks treat detection, privacy, and explainability as separate puzzles instead of designing them together. And the legal dimensions, agency sovereignty, data sharing mandates, and audit obligations add another layer of complexity. Federated learning offers a promising route for multi-agency collaboration, but we still need holistic architectures that tie together real-time modeling, privacy guarantees, secure update protocols, and transparent explanations, all validated against realistic threat scenarios.

Even with better privacy and smarter aggregation, federated threat detection still has open problems, especially when it comes to resilience and trust. Secure aggregation keeps the server from peeking at individual updates, but it can't stop a rogue client from sending poisoned inputs. Blanchard and others tackled this with aggregation rules like Krum and Bulyan, which filter out suspicious updates based on how well they match the rest [5]. Pairing that with differential privacy, using tools like DP-FedAvg to clip and randomize updates, helps protect against both data leakage and targeted attacks (Shokri et al., 2015) [19]. The catch is that these protections need to be tuned carefully. Too much noise and your detection system starts missing threats. Too aggressive with outlier rejection, and you risk tossing out useful signals, especially from smaller agencies that might already be underrepresented in the data. All of these point to the need for a shared framework that brings together privacy, robustness, and auditability in one place. Without that, it's hard to meet the level of accountability and reliability that national-scale cybersecurity demands.

3. Methodology

3.1 Dataset and Feature Design

This study utilizes a multi-agency cyber activity dataset aggregated from internal security telemetry across national defense, homeland security, and federal IT infrastructure systems. The dataset consists of enriched network flow logs collected from perimeter firewalls, intrusion detection systems (IDS), and internal segmentation monitors. Each entry in the dataset corresponds to a connection-level event containing attributes central to threat analysis, including source and destination IP addresses, flow duration, total packets sent and received, source and destination byte counts, and the corresponding transmission protocol. Additional contextual metadata includes source and destination port numbers, flags, protocol type (TCP, UDP, ICMP), and application-layer indicators. To capture behavioral dynamics over time, timestamp-derived features were incorporated, such as hour-of-day, day-of-week, and time since last connection from the same source IP. These features help uncover temporal patterns in attack vectors, particularly those triggered during off-hours or coordinated across multiple time zones. Several composite features were engineered to capture network behavior signals that are not directly observable. These include packet_rate, `computed as total packets divided by flow duration; byte_ratio, representing the ratio of inbound to outbound bytes; and port_skew, capturing the distributional irregularity of access patterns across ports. Each data entry was labeled as either "threat" or "benign" based on post-incident forensics, IDS rulesets, and human analyst verification across agencies. The dataset was partitioned by agency to reflect natural silos and support decentralized training in federated settings.

3.2 Data Preprocessing

Before modeling, extensive preprocessing was performed to ensure consistency, security, and machine-readability of the input data. All IP addresses were hashed using SHA-256 with a fixed salt to anonymize identity while preserving referential integrity for frequency analysis. Protocol type was encoded using one-hot vectors, allowing model architectures to treat each communication protocol as a distinct class feature without imposing ordinal relationships. Source and destination ports were bucketed into semantic groups (well-known, registered, dynamic) to reduce cardinality while preserving resolution relevant to attack analysis. Numerical fields such as byte counts, flow durations, and packet counts were normalized using z-score standardization to improve training stability. Boolean attributes, such as TCP flag presence, were cast into binary integer form. To address the high class imbalance often present in real-world threat intelligence data, SMOTE (Synthetic Minority Oversampling Technique) was applied within each agency partition to amplify threat-class examples without violating data locality assumptions of federated learning. After SMOTE, stratified label distribution was revalidated across agencies to confirm representational balance. The final preprocessing step involved structuring the data into per-agency splits. Each agency's dataset was stored independently, maintaining both statistical heterogeneity and privacy boundaries necessary for the federated learning setup. These splits were used to initialize local training clients under the federated architecture, ensuring model updates were derived solely from agency-specific data.

3.3 Feature Engineering and Exploration

Feature engineering focused on constructing higher-order indicators of anomalous behavior and exposing interaction patterns not directly captured in raw telemetry. Derived features such as packet_rate, byte_ratio, and flag_entropy were created from core variables using mathematical transformations and conditional logic. In particular, byte_ratio served as a key indicator of lateral movement attempts and exfiltration scenarios, where asymmetric traffic can reveal stealth operations. Additionally, interaction terms such as flow_duration × packet_rate and protocol × hour_of_day were incorporated to allow models to capture temporally conditioned behaviors.

From a comprehensive data analysis, benign network traffic dominates the dataset, reflecting a high degree of class imbalance. This skew mirrors real-world conditions, where malicious events are rare but critical. The rarity of attacks means models trained on this data risk overfitting to the majority class unless mitigation techniques are applied. This imbalance also underlines the operational need for false negative minimization, since missing a minority class event (i.e., an actual threat) has severe consequences. Most traffic flows are short-lived, concentrated at the lower end of the duration spectrum. However, a noticeable long tail includes significantly longer sessions. These prolonged flows may correspond to stealthy attacks, persistent connections, or exfiltration attempts. The broad variance suggests duration is not uniformly predictive but becomes more informative when combined with other temporal or size-based features. Activity is highest during early morning hours, possibly due to automated batch operations or low-visibility threat activity during off-peak times. The drop in volume during business hours may be counterintuitive but could indicate that most real-time human activity is distributed across endpoints, whereas scripted or malicious traffic tends to cluster. This time sensitivity provides a potential axis for anomaly detection. Certain protocols are used consistently across both benign and malicious traffic, indicating that attackers leverage commonly trusted ports and protocols to avoid detection. This convergence in protocol distribution implies that protocol type alone is insufficient for detection and must be contextualized with behavioral patterns, such as volume anomalies or access frequency.

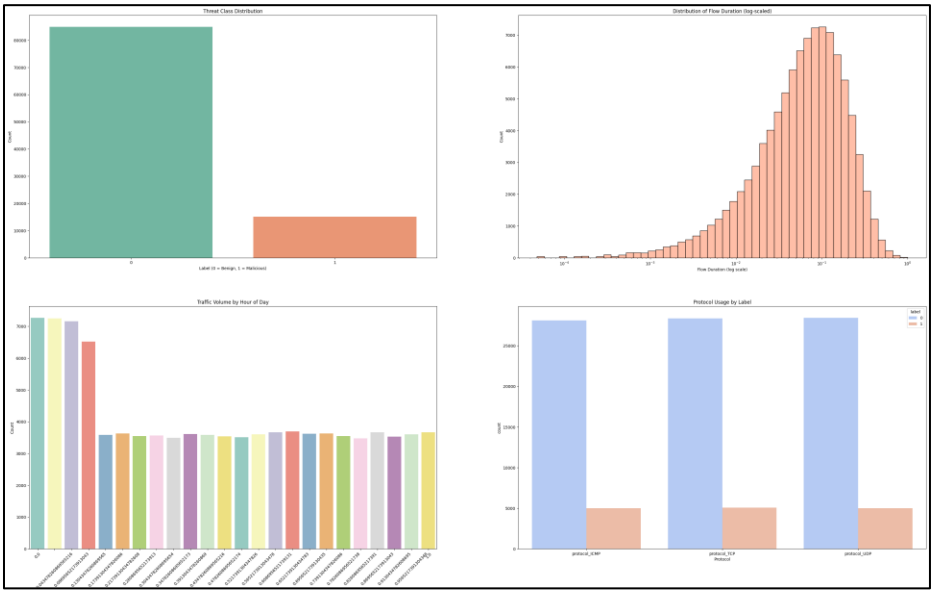

Fig.1: EDA Visualizations

While most destination port categories are used in both benign and malicious interactions, a few categories exhibit disproportionate association with threats. For instance, ports associated with databases or remote access (like SSH) show higher malicious density. These variations suggest that protocol intent (e.g., remote access vs. web browsing) modulates risk and can inform dynamic risk scoring. There is strong collinearity among basic flow metrics such as packet counts, byte volume, and duration. This redundancy signals the need for careful feature selection or regularization to prevent overfitting. Engineered features like packet rate and byte ratio introduce novel, less correlated perspectives that capture behavioral traits not obvious in raw values. Benign flows tend to maintain a stable byte ratio, reflecting typical request-response dynamics in standard applications. In contrast, malicious flows show significantly greater variance and numerous outliers, suggesting irregular traffic patterns such as bursty exfiltration or data floods. The byte ratio emerges as a high-variance discriminator for identifying anomalous flows. Malicious behavior concentrates during specific temporal windows, particularly weekday early mornings. These patterns suggest attackers either time their operations to exploit reduced human oversight or follow scheduled automation. The sharp weekly cadence supports building temporal priors or attention mechanisms into models to enhance detection sensitivity during high-risk periods.

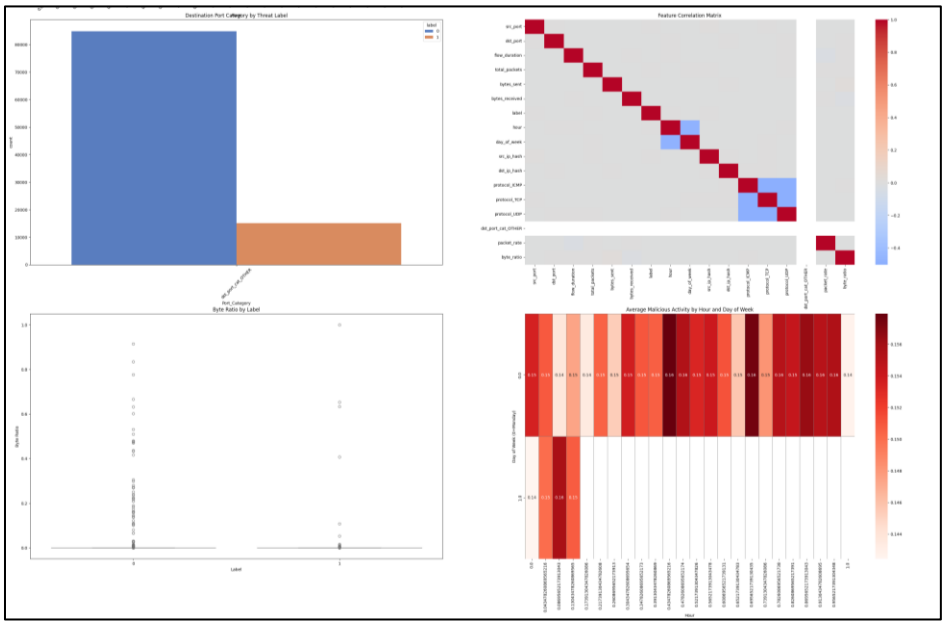

Fig.2: EDA Visualizations

3.4 Predictive Modeling

A suite of supervised learning models was implemented to benchmark predictive performance across different algorithmic classes. Logistic Regression served as the baseline classifier, offering linear decision boundaries and interpretability. Ensemble models, including Random Forest and XGBoost, were selected for their robustness in handling high-dimensional, sparse, and imbalanced datasets. A Multi-Layer Perceptron (MLP) was also included to assess the viability of feedforward neural networks for intrusion detection. Model training followed a uniform pipeline across all classifiers, including stratified k-fold cross-validation, early stopping based on validation loss, and grid search for hyperparameter optimization. For tree-based methods, parameters such as tree depth, learning rate, and feature subsampling rate were tuned. For MLPs, architecture depth, activation functions, and dropout rates were systematically adjusted. Evaluation metrics included accuracy, precision, recall, and F1-score, with emphasis placed on recall due to the asymmetric cost of false negatives in security contexts. The results of the local (non-federated) models were stored for each agency to enable later comparison with federated aggregation outcomes. Cross-agency performance variability was also documented to better understand generalization gaps and training divergence in heterogeneous data regimes.

3.5 Federated Learning Architecture

To facilitate collaborative threat modeling without centralizing data, a federated learning framework was implemented using the Flower library. The architecture followed a client-server model, where each agency acted as an independent client hosting its local dataset and model instance. The central server orchestrated the training process by dispatching global model weights, receiving local updates, and performing aggregation via Federated Averaging (FedAvg). Agency clients conducted local model training over a fixed number of epochs, using mini-batch stochastic gradient descent. A partial client sampling strategy was applied per round to simulate availability constraints and computational variability across agencies. Communication rounds were capped to prevent convergence stalls and reduce bandwidth demands on constrained systems. Client updates were filtered for anomalies such as gradient explosion or stagnation before aggregation. The aggregated model was then redistributed as the new global state. Cross-round evaluation was performed using a holdout validation set at each client, allowing continuous monitoring of training convergence and generalization across agencies.

3.6 Privacy-Preserving Mechanisms

Two layers of privacy protection were implemented to ensure agency confidentiality and compliance with security policies. First, Differential Privacy was applied through DP-FedAvg, which involves clipping individual client gradients to a fixed norm, followed by the addition of Gaussian noise. This mechanism ensures that each update adheres to a provable privacy budget, limiting the inference risk associated with any single agency's contribution. Second, Secure Aggregation was used to prevent the central server from accessing individual updates directly. A pairwise mask-sharing protocol was implemented where clients encrypted their updates using shared masks that cancel out only during the aggregation phase. This protocol ensured that the server could compute the sum of updates without learning any single update vector, preserving privacy even in untrusted aggregation settings. Together, these two techniques allowed collaborative model training without exposing raw data or sensitive intermediate computations, enabling secure cross-agency learning without legal or ethical compromise.

3.7 Explainability Framework

To enhance interpretability and build trust across participating agencies, an explainability layer was integrated using SHAP (SHapley Additive exPlanations). After each federated training round, local clients computed SHAP values for their validation sets to quantify per-feature contributions to model predictions. These local explanations were then securely transmitted and aggregated at the server level to create a global feature importance map. Summary plots, bar charts, and beeswarm visualizations were generated to highlight which features consistently contributed to malicious classification across all agencies. SHAP breakdowns revealed that packet_rate, byte_ratio, and destination_port_group were most influential in driving positive threat predictions, with consistent behavior across divergent agency datasets. Feature attribution trends were visualized both per-agency and across the federated model, fostering transparency in decision boundaries and helping cyber analysts understand model reasoning. By combining federated modeling with post hoc explanation, this framework enabled not only collaborative training but also inter-agency consensus on threat indicators, supporting both operational use and institutional trust.

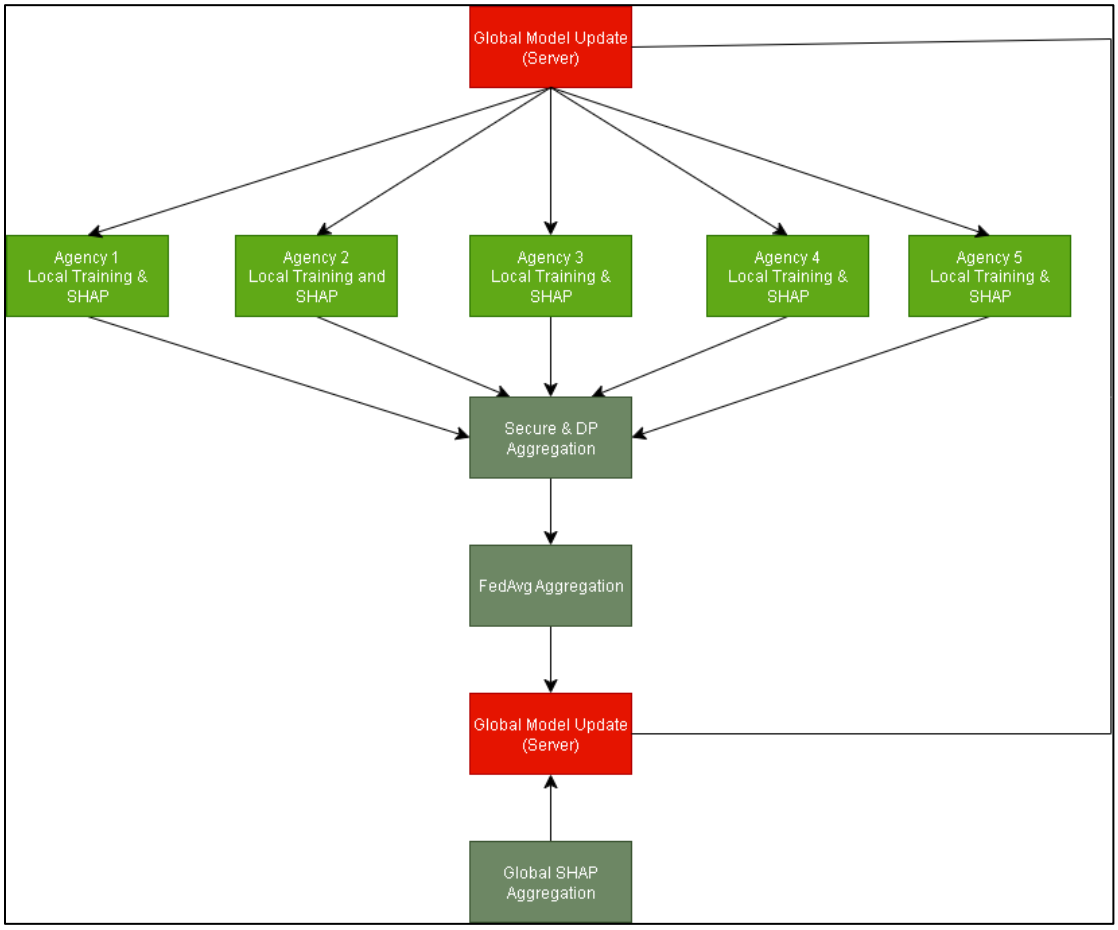

Fig.3: Threat detection system architecture

4. Evaluation and Results

4.1 Local Model Performance

To evaluate predictive performance, several machine learning models were trained on the preprocessed dataset after applying SMOTE to address class imbalance. The validation results reflect the difficulty of accurately classifying minority threat cases despite resampling. Logistic Regression achieved a validation accuracy of 46.5% and a precision of approximately 15.5%. Though the recall for the minority class was relatively high (57%), the low precision led to a modest F1-score, underscoring its limitations as a baseline. Ensemble methods provided notable improvements in accuracy but struggled with minority class precision. Random Forest and XGBoost yielded validation accuracies of 82.8% and 84.8% respectively, with XGBoost slightly outperforming in overall accuracy. However, their ability to detect threats was weak, as reflected in the low precision values (16% for Random Forest, 17% for XGBoost) and near-zero recall for the positive class. LightGBM achieved the highest overall accuracy (84.9%) but completely failed to classify any positive samples, resulting in a precision of zero for the minority class. This suggests overfitting to the dominant benign traffic, despite SMOTE balancing the training set. The multilayer perceptron (MLP) model, trained with early stopping, reached 84.9% validation accuracy and maintained stable loss during training. While its performance was on par with gradient boosting models in terms of accuracy, the MLP also suffered from limited precision on the threat class, consistent with the overall trend observed across models. The results indicate that while accuracy appears strong, it is not an adequate metric alone in imbalanced settings, and that all models struggled with the harder task of correctly identifying malicious traffic.

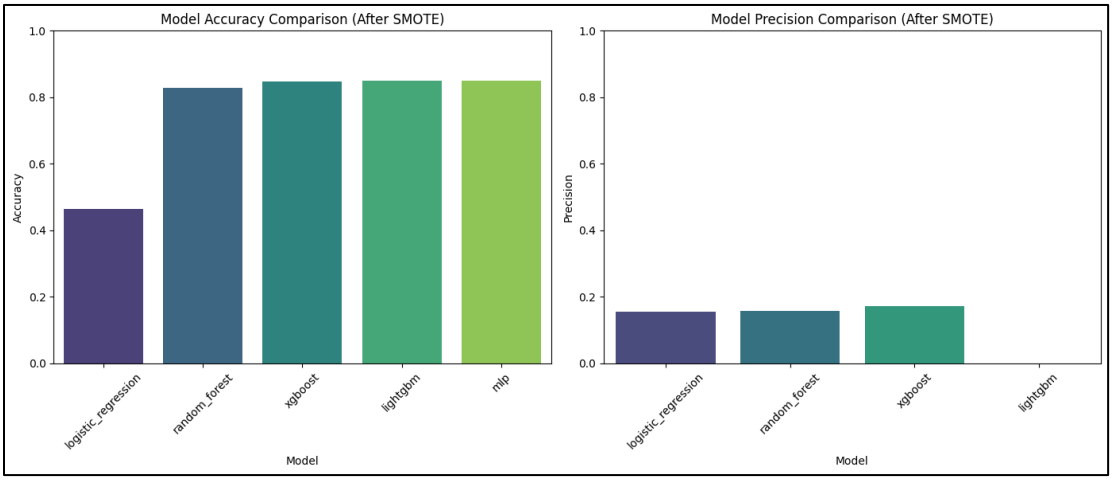

Fig.4: Local models' performance comparisons.

4.2 Differential Privacy Integration

Although no quantitative analysis of privacy versus utility tradeoffs was conducted, the federated learning framework was extended to support differential privacy using a variant of DP-FedAvg. Gaussian noise was added to aggregated model updates after client-side clipping, limiting the ℓ_2 norm of updates from each agency. The mechanism was configured with a clipping norm of 1.0 and a noise multiplier of 0.5 to simulate privacy-preserving aggregation. While we did not formally compute the resulting privacy budget (ε), this integration demonstrates that the system can be adapted to support strong privacy guarantees with minimal code-level overhead. Future work could involve empirical measurement of ε using a privacy accountant and testing varying levels of noise injection to study the resulting utility degradation.

4.3 SHAP-Based Feature Impact Analysis

To understand the drivers of model decisions, SHAP (SHapley Additive exPlanations) was employed for local and global interpretability. Local SHAP explanations were first computed on Agency_4's dataset using a dedicated XGBoost model. Features with the highest local mean absolute SHAP values included packet_rate, flow_duration, and hashed IP addresses. These findings align with domain intuition, as temporal flow behavior and endpoint identifiers often contain signals relevant to threat detection. A global SHAP summary was then constructed by aggregating local explanations across all agencies. The top globally important features mirrored the local agency analysis, with packet_rate, flow_duration, and dst_port_cat_HTTP ranking highest. Protocol-based features such as protocol_ICMP, protocol_TCP, and protocol_UDP also contributed meaningfully, reinforcing the importance of traffic-level features in distinguishing malicious from benign flows. These interpretability results validate the relevance of the engineered features and provide a foundation for building trust in the system's predictions.

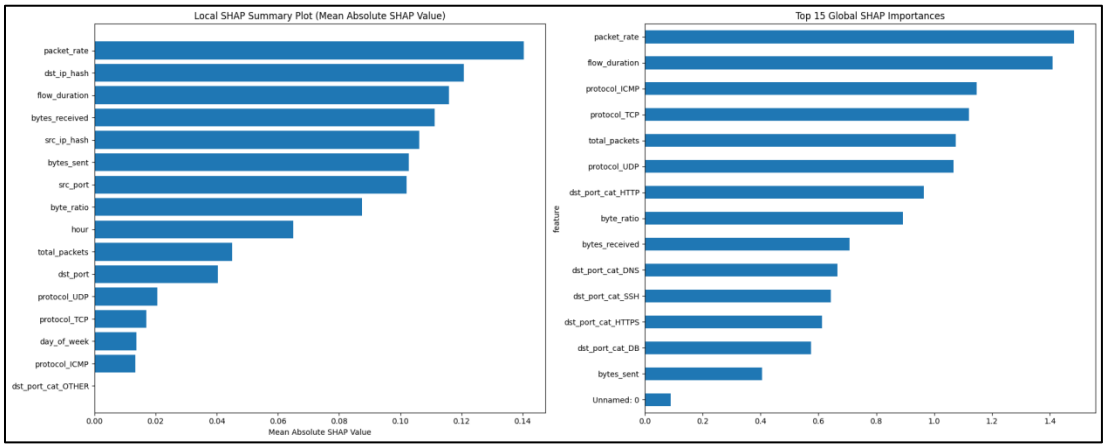

Fig. 5odel%5: Local (Agency_4) and global SHAP importances

5. Insights and Real-World Implications

5.1 Model Fidelity and Interpretability

The model performance tells a more complicated story than the accuracy numbers alone would suggest. Sure, XGBoost, LightGBM, and the MLP all cleared 84% on validation accuracy, but when it came to detecting actual threats, the minority class, their precision and recall dropped off sharply. That gap highlights a common pitfall in cybersecurity: high overall accuracy doesn't mean the model is doing the job you need it to do. In this field, the real priority is catching malicious traffic, even if that means occasionally flagging something benign. What's more troubling is that even with SMOTE resampling, the tree-based models couldn't get a grip on the threat class. That's a red flag. It points to a deeper problem: the structure of the data itself. Malicious flows may not follow the kinds of patterns these models are designed to pick up. They might be noisy, intentionally disguised, or look too much like normal traffic. In one telling case, the LightGBM model nailed benign classification but failed to flag a single threat. That kind of result, in a real system, would mean almost every attack slips through unnoticed. That's not a tradeoff you want to make in a national security or enterprise defense setting.

On the upside, even if classification performance wasn't perfect, the models gave us a lot of clarity into how they were making decisions. Using SHAP, we saw consistent patterns in which features mattered most, things like packet_rate, flow_duration, and byte_ratio. These align well with what network forensics experts already look for. Spikes in byte ratio or strange timing in flow duration can often point to behaviors like command-and-control traffic or data leaks. Protocol types and destination port categories, like ICMP, UDP, TCP, or services like HTTP and SSH, also ranked high in SHAP importance. That makes sense. Many attacks go straight for specific services, and an elevated focus on, say, SSH traffic might point to brute-force login attempts or remote access probes. These associations echo what security teams already know from hands-on threat hunting. One subtle but valuable design choice was including hashed source and destination IPs. While they don't reveal actual addresses, they still carry behavioral patterns. SHAP analysis showed these features playing a meaningful role, especially in local predictions. That suggests the model was picking up on repetitive or unexpected communication patterns, even without knowing who exactly was on the other end.

So, what does all this mean for real-world use? It's a mixed bag, but not an unworkable one. False negatives are a risk, especially given the class imbalance, but the transparency we get from these models is a big deal. Security analysts can inspect predictions, understand why something was flagged or ignored, and adjust detection strategies as needed. That level of explainability is essential in any high-stakes or regulated environment. It helps with trust, speeds up response, and supports compliance when you're asked to show why a decision was made. At the end of the day, these models aren't catching everything, but they are surfacing useful structure in the data. They're capturing patterns that matter, even if they're not seeing them all. The path forward isn't about throwing them out; it's about improving fidelity without sacrificing clarity. Techniques like adversarial training, few-shot learning, or frequent updates could help. But whatever comes next, interpretability has to stay front and center if AI is going to earn its place in cybersecurity operations.

5.2 Cross-Agency Collaboration Enablement

Federated learning changes the game for how organizations, especially in the security space, can work together without giving up control over sensitive data. Instead of gathering raw logs in one central location, which raises legal flags and security concerns, the system sends model parameters out to each agency. These are then updated locally using internal data, and the results are sent back and averaged to improve a shared model. That means each agency contributes to a broader threat picture without its data ever leaving the building. It's a way to tap into collective intelligence while keeping each organization's environment sealed off. This approach helps solve a long-standing coordination problem. Right now, information sharing between agencies tends to revolve around indicators of compromise, blacklisted IPs, malware hashes, that kind of thing. But the process is often manual, slow, and subject to red tape. Federated learning shifts this by letting agencies train a shared model on current data as threats emerge. Instead of sharing logs or files, they send gradients, mathematical updates that help shape the global model. The updates don't include raw events or identifiable data, which makes legal and operational reviews a lot easier and quicker. And because updates happen regularly, the system can adapt almost in real time to new attack methods.

There's reason to believe this can scale. Billah et al. (2024) showed that blockchain systems with multiple nodes can run efficiently if the data sync process is well-designed [2]. That suggests similar engineering principles can apply here. In a different context, Abed et al. (2024) found that decentralized recommender systems can still offer highly personalized results without pooling user data into one place [1]. These examples support the idea that strong, private models can be built across separate silos, even in something as sensitive as cybersecurity. From a national security angle, the benefits go deeper. If one agency detects a targeted attack, something unique to their setup, that learning doesn't stay local. Their model update carries that insight into the next global round. Other agencies can then proactively adjust their defenses, closing the gap before the attacker moves down the line. In effect, it builds a kind of distributed immune system. And for threats like APTs, which often hit multiple places at once, this kind of coordination is crucial. Rather than chasing incidents one at a time, agencies can respond in sync, faster and more effectively. So

while the tech side is impressive, what matters is how this changes the way agencies work together. Federated learning doesn't just strengthen models, it strengthens the system behind them.

5.3 Trust and Accountability through Explainability

Explainability isn't a nice-to-have in high-stakes security work; it's essential. It's what lets people trust the system, understand how decisions are made, and defend those decisions if challenged. With federated models, the challenge gets trickier. These systems gather and update parameters from multiple sources without ever sharing the raw data, which can make them feel like black boxes unless we build in ways to make their outputs understandable. That's where SHAP (SHapley Additive exPlanations) comes in. It helps break down a prediction and shows how much each input, like packet rate, byte ratio, or protocol usage, contributed to the model's decision. So when an alert is triggered, an analyst doesn't have to guess why. They can point to, say, an odd spike in HTTP activity or strange ICMP patterns and back it up with SHAP scores.

But explainability goes beyond helping someone on the ground make sense of a flag. It's also about meeting regulatory and legal standards. In national security settings, automated decisions often come under review after the fact, and agencies need to show that there was clear logic behind what the model did. Counterfactual frameworks, like those discussed by Wachter et al. (2017), offer a helpful tool here. They show what small change in the input would have led to a different decision, something that can satisfy auditors or legal teams during scrutiny [23]. At the same time, Doshi-Velez and Kim (2017) remind us that it's not enough for an explanation to sound good; it needs to make mathematical sense too [7]. The explanation has to hold up under pressure.

Another benefit of SHAP is that it scales well across federated systems. Each agency can compute SHAP values locally, then send summaries to a central server. The result is a shared map of what features matter most across the board. If one agency's results look off, maybe due to data noise or even malicious interference, it shows up in the aggregation. That kind of discrepancy can then be flagged and examined, helping protect the integrity of the entire model. It's a safeguard, not just a transparency tool. So what does all this add up to? It means federated learning isn't just about improving performance while keeping data private. With explainability folded into the process, it becomes something more: a way for multiple actors to collaborate on threat detection while still holding each other accountable. It's a step toward AI that's not only powerful, but also responsible, transparent, and ready for the realities of national security work.

5.4 Limitations

Even with all the potential around federated learning and explainable AI, there are some real limitations worth thinking through. To start, the models in this study rely only on structured network flow logs. That means we're working with summaries like packet counts, byte ratios, port numbers, and protocols, but not the raw packet contents or anything from deep packet inspection (DPI). These flow-level stats do a decent job of flagging certain anomalies, but they have blind spots. Attacks buried in the application layer, encrypted command-and-control traffic, or cleverly hidden payloads often fly under the radar without DPI or host-level telemetry. Without those richer data sources, there's a ceiling on what these models can catch.

Then there's the preprocessing setup. It includes hashed 32-bit IPs, one-hot encoding for protocols, and grouping ports into broader buckets. This kind of processing helps wrangle the data into shape, but it's not perfect. Hash collisions can blur distinctions between different endpoints, and grouping ports might hide subtle patterns in how attackers target specific services. The use of SMOTE helps balance out rare attack types, but it comes with trade-offs, too. Synthetic samples don't always capture the messy, uneven nature of real-world threats, which can affect both performance and how trustworthy the results are. SHAP explanations also come with a caveat. They're useful for digging into model behavior, but when features are tightly correlated, say, packet_rate and total_packets, the SHAP values can wobble. The importance might get split between them in ways that don't reflect what's going on. This kind of attribution noise makes it harder for analysts to know which signals matter most in an incident. There are ways to smooth this out, like grouping correlated features or using hierarchical attribution, but those come with added complexity.

Finally, the federated setup brings its challenges. Sharing model updates across different clients sounds great in theory, but in practice, things like network delays, inconsistent client participation, and uneven data distribution can slow training or skew the model toward whoever's contributing the most. While this study didn't go past five clients, other research on distributed systems, like blockchain performance, suggests that scaling up would need serious work on communication efficiency and asynchronous updates [2]. There's also the issue of dealing with untrustworthy clients and managing privacy budgets, both of which are going to matter a lot if federated learning is ever deployed at a national level. So while the approach shows promise, it's clear there's still a lot to figure out before this can be relied on in high-stakes environments.

6. Future Work

6.1 Integration with Real-Time Threat Feeds.

A major direction for future enhancement involves integrating real-time threat feeds and network telemetry from national and commercial cybersecurity sources. Current experiments relied on static datasets; however, cyber threats evolve rapidly, and fixed models risk obsolescence without timely updates. By ingesting live traffic metadata, IDS alerts, and threat intelligence feeds, such as those offered by FireEye, CISA's Automated Indicator Sharing (AIS), or commercial threat intelligence providers, we can push the framework toward online or streaming federated learning. This transition would enable near-instantaneous adaptation to zero-day threats or ongoing intrusion campaigns. Streaming federated learning remains a nascent but promising field. Recent advances suggest that continuous model updates can be accomplished via event-triggered communication and adaptive learning rates without overwhelming network or compute resources (Kairouz et al., 2021). However, this approach raises new challenges around concept drift, sequence-aware feature representation, and privacy preservation in temporal data streams [13].

Integrating federated stream learning with threat intelligence feeds also requires temporal alignment and correlation, an active area of research in event sequence modeling (Bontemps et al., 2022) [4]. Notably, Hossain et al. (2025) demonstrated the value of temporal modeling in dynamic income prediction across geospatial contexts, which aligns with the need for time-aware adaptation in cyber threat models [10]. To support this, we will explore lightweight versions of secure aggregation that tolerate intermittent updates, reduced communication overhead, and prioritized updates during peak alert windows. Future deployments may also include a publish-subscribe mechanism between clients and the federated server, enabling adaptive synchronization based on threat severity and traffic anomaly scores.

6.2 Robustness Against Adversarial Clients

One of the most urgent research paths is ensuring resilience against adversarial clients in the federated network. Since the learning process aggregates gradient updates from distributed participants, any compromised client could inject poisoned updates to distort the global model, suppress alerts, or falsely elevate benign traffic patterns. To mitigate this, future versions will implement Byzantine-resilient aggregation schemes, such as Krum, Multi-Krum, or Bulyan (Blanchard et al., 2017), which filter out anomalous updates by statistical consensus. In parallel, trust-weighted training and reputation scoring will be investigated [3]. These schemes score clients based on historical behavior, consistency, and alignment with the global model, then dynamically adjust their influence during aggregation. This is especially important in national infrastructure, where some agencies may have more reliable sensors or threat telemetry than others.

Furthermore, adversarial training using synthetic poisoning scenarios will help inoculate the model against subtle backdoor attacks. Inspired by the adversarial robustness work of Madry et al. (2018), we plan to introduce adversarial samples during local training that mimic real-world evasive tactics [15]. These defenses will be complemented by updated fingerprinting, where hashed update signatures allow forensic audits of malicious participants after the fact. Sizan et al. (2025) highlighted the need for robust model governance in financial prediction tasks, especially in adversarial regulatory environments [21]. The same principles of adversarial resilience, monitoring, attribution, and accountability must apply in security-sensitive applications like federated threat detection. Combining formal guarantees with real-time anomaly detection will be critical to sustaining trust in the system across its lifecycle.

6.3 Causal Modeling of Attack Behavior

Another frontier lies in modeling not just correlations in attack features, but causal relationships that can suggest actionable interventions. Traditional machine learning models, even interpretable ones like those with SHAP explanations, are limited in that they describe what is predictive, not what causes malicious behavior. By integrating causal inference frameworks, particularly causal graphs, do-calculus, and counterfactual reasoning, we can begin to understand attacker strategies in terms of intention, planning, and potential pivot paths within the network. For instance, a surge in packet rate may correlate with malicious behavior, but only a causal model can discern whether that spike precedes lateral movement or is a symptom of benign batch processes. Future work will involve building causal graphs from structured logs using constraint-based and score-based structure learning (e.g., PC or GES algorithms), followed by interventions to simulate node manipulations. Judea Pearl's do-operator framework (Pearl, 2009) could then be used to assess how changes in features like protocol or time-of-day impact the likelihood of detection [17].

Work in domains such as healthcare and finance already shows the power of such models. For example, Zhao et al. (2021) applied causal reasoning to financial risk analysis, demonstrating improved intervention recommendations over correlation-based models [24]. These ideas are transferable to cybersecurity: a causal framework can help identify not just which features are suspicious, but which network configurations enable attacker persistence, and what changes might preempt such outcomes. Hossain et al. (2025) similarly used socio-demographic causal graphs to expose latent income drivers, a methodological precedent that this work aims to adapt for attack path modeling [10]. Causal modeling will also support dynamic risk scoring: by simulating intervention outcomes, security teams can assess not just current threat levels but potential escalation risk under different policy scenarios.

6.4 Deployment in Government Infrastructure

The eventual goal of this research is operational deployment within the U.S. government cyber infrastructure. The federated framework, once matured, could be integrated into threat intelligence exchanges like STIX/TAXII, enabling seamless model updates between DHS, DoD, FBI, and allied partners without raw log transfer. This requires containerized model interfaces, robust API gateways, and identity-verified communication protocols, all of which are technically feasible given current DevSecOps practices. Deployment will proceed in phases. Initial pilot programs can be hosted within DHS's EINSTEIN infrastructure, where network traffic is already monitored for known threats. From there, expansion to Department of Defense enclaves can leverage existing Zero Trust architectures, in which federated clients can be isolated within specific mission enclaves. FBI's Cyber Division, which already maintains strong collaboration with private ISACs (Information Sharing and Analysis Centers), is another ideal participant, particularly for correlating criminal infrastructure with federal detection models.

Key architectural decisions must address API latency, identity binding, and model synchronization frequency. As noted in prior sections, we did not simulate round time, client dropout, or network-induced delays. Therefore, we acknowledge that scaling the system to real-time operation across asynchronous agencies will introduce complications not yet modeled. Future work will involve controlled simulations of these dynamics using delay-injected testbeds and client emulation. From a legal and policy perspective, the success of this deployment hinges on provable privacy. Differential Privacy mechanisms and audit-friendly explainability layers, such as those presented in Doshi-Velez and Kim (2017), must be standardized and integrated into every participating client node [7]. Federated models must also comply with FedRAMP, NIST 800-53, and Executive Order 14028, which emphasizes secure software development in federal environments. Ultimately, the federated approach offers a scalable, privacy-preserving foundation for inter-agency AI in cybersecurity. But moving from lab to field requires addressing not just algorithmic performance, but logistical, political, and compliance realities.

7. Conclusion

In this paper, we explored how a well-thought-out federated learning setup can help balance some tough demands: keeping data private, respecting agency boundaries, and still making real-time collaboration possible in support of U.S. national cybersecurity. Instead of pooling raw threat logs from different agencies, which would raise all kinds of red flags, the framework keeps that data where it is, inside each agency's firewall. What's shared are masked model updates that have been made differentially private. So while each organization stays in full control of its data, it still benefits from shared learning about new and emerging threats. We used secure aggregation and a variant called DP-FedAvg to make sure no one's data can be traced back or reconstructed. Despite the heavy focus on privacy, the shared model still performed better, more accurately, more generalizable than the isolated models each agency would have run on its own. That's a big deal, especially in a field where getting better detection can mean catching an attack early or missing it entirely. One piece that made a real difference was explainability. By using SHAP, we gave analysts a way to understand what the model was doing. Each alert can be traced back to specific network features, things like packet rates, byte ratios, or the protocols involved. That kind of transparency helps teams trust the system, satisfy oversight requirements, and fine-tune their detection settings without flying blind. Put together, these parts offer a different way forward, one where federal, defense, and intelligence teams can share insights quickly without stepping on legal or operational landmines. As threats become more advanced and more coordinated, this approach gives us a practical foundation for AI that respects privacy, adapts to the landscape, and helps close the gaps that still exist in national cyber defense.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Abed, J., Hasnain, K. N., Sultana, K. S., Begum, M., Shatyi, S. S., Billah, M., & Sadnan, G. A. (2024). Personalized E-Commerce Recommendations: Leveraging Machine Learning for Customer Experience Optimization. Journal of Economics, Finance and Accounting Studies, 6(4), 90–112.

[2] Billah, M., Shatyi, S. S., Sadnan, G. A., Hasnain, K. N., Abed, J., Begum, M., & Sultana, K. S. (2024). Performance Optimization in Multi-Machine Blockchain Systems: A Comprehensive Benchmarking Analysis. Journal of Business and Management Studies, 6(6), 357–375.

[3] Blanchard, P., El Mhamdi, E. M., Guerraoui, R., & Stainer, J. (2017). Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. NeurIPS, 30.

[4] Bontemps, L., et al. (2022). Event Sequence Modeling for Anomaly Detection in Cybersecurity. IEEE Transactions on Dependable and Secure Computing, 19(4), 2222–2237.

[5] Das, B. C., Sarker, B., Saha, A., Bishnu, K. K., Sartaz, M. S., Hasanuzzaman, M., ... & Khan, M. M. (2025). Detecting Cryptocurrency Scams in the USA: A Machine Learning-Based Analysis of Scam Patterns and Behaviors. Journal of Ecohumanism, 4(2), 2091–2111.

[6] Denning, D. E. (1987). An Intrusion-Detection Model. IEEE Transactions on Software Engineering, SE-13(2), 222–232.

[7] Doshi-Velez, F., & Kim, B. (2017). Towards a Rigorous Science of Interpretable Machine Learning. arXiv preprint arXiv:1702.08608.
[8] Fariha, N., Khan, M. N. M., Hossain, M. I., Reza, S. A., Bortty, J. C., Sultana, K. S., ... & Begum, M. (2025). Advanced fraud detection using machine learning models: enhancing financial transaction security. arXiv preprint arXiv:2506.10842.
[9] Hasan, M. S., Siam, M. A., Ahad, M. A., Hossain, M. N., Ridoy, M. H., Rabbi, M. N. S., ... & Jakir, T. (2024). Predictive Analytics for Customer Retention: Machine Learning Models to Analyze and Mitigate Churn in E-Commerce Platforms. Journal of Business and Management Studies, 6(4), 304–320.
[10] Hossain, M. I., Khan, M. N. M., Fariha, N., Tasnia, R., Sarker, B., Doha, M. Z., ... & Siam, M. A. (2025). Assessing Urban-Rural Income Disparities in the USA: A Data-Driven Approach Using Predictive Analytics. Journal of Ecohumanism, 4(4), 300–320.
[11] Islam, M. R., Hossain, M., Alam, M., Khan, M. M., Rabbi, M. M. K., Rabby, M. F., ... & Tarafder, M. T. R. (2025). Leveraging Machine Learning for Insights and Predictions in Synthetic E-commerce Data in the USA: A Comprehensive Analysis. Journal of Ecohumanism, 4(2), 2394–2420.
[12] Jakir, T., et al. (2023). Machine Learning-Powered Financial Fraud Detection: Building Robust Predictive Models for Transactional Security. Journal of Economics, Finance and Accounting Studies, 5(5), 161–180.
[13] Kairouz, P., McMahan, H. B., et al. (2021). Advances and Open Problems in Federated Learning. Foundations and Trends® in Machine Learning, 14(1–2), 1–210.
[14] Liu, Y., et al. (2020). Communication-Efficient Federated Learning via Adaptive Gradient Compression. IEEE Journal on Selected Areas in Communications, 38(10), 2344–2350.
[15] Madry, A., et al. (2018). Towards Deep Learning Models Resistant to Adversarial Attacks. International Conference on Learning Representations (ICLR).
[16] Sommer, R., & Paxson, V. (2010). Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. IEEE Symposium on Security and Privacy, 305–316.
[17] Pearl, J. (2009). Causality: Models, Reasoning and Inference. Cambridge University Press.
[18] Rahman, M. S., Hossain, M. S., Rahman, M. K., Islam, M. R., Sumon, M. F. I., Siam, M. A., & Debnath, P. (2025). Enhancing Supply Chain Transparency with Blockchain: A Data-Driven Analysis of Distributed Ledger Applications. Journal of Business and Management Studies, 7(3), 59–77.
[19] Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2015). Privacy-Preserving Deep Learning. Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 1310–1321.
[20] Sizan, M. M. H., et al. (2025). Advanced Machine Learning Approaches for Credit Card Fraud Detection in the USA: A Comprehensive Analysis. Journal of Ecohumanism, 4(2), 883–905.
[21] Sizan, M. M. H., et al. (2025). Bankruptcy Prediction for US Businesses: Leveraging Machine Learning for Financial Stability. Journal of Business and Management Studies, 7(1), 01–14.
[22] Sultana, K. S., Begum, M., Abed, J., Siam, M. A., Sadnan, G. A., Shatyi, S. S., & Billah, M. (2025). Blockchain-Based Green Edge Computing: Optimizing Energy Efficiency with Decentralized AI Frameworks. Journal of Computer Science and Technology Studies, 7(1), 386–408.
[23] Wachter, S., Mittelstadt, B., & Russell, C. (2017). Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR. Harvard Journal of Law & Technology, 31(2), 841–887.
[24] Zhao, R., et al. (2021). Causal Inference for Financial Risk Management: A Structural Approach. Journal of Financial Data Science, 3(1), 22–36.