LEX LOCALIS-JOURNAL OF LOCAL SELF-GOVERNMENT
ISSN:1581-5374 E-ISSN:1855-363X
VOL. 24, NO. 4 (2026)

AN EXPLAINABLE MACHINE LEARNING FRAMEWORK FOR CYBERSECURITY AND DIGITAL SELF GOVERNANCE IN LOCAL GOVERNMENT NETWORKS

R Kalaivani[1*], Dr. V.Vennila [2]

[1*] Assistant Professor, Department of Cyber Security, Mahendra Engineering College, Namakkal, India
[2]Professor, Department of Computer Science and Engineering, Nandha College of Technology, Erode, India
Corresponding Author: kalaivani10@gmail.com

Abstract-Local governments increasingly depend on connected service platforms, networked sensors, digital records, and inter-organizational data exchange. These systems create an operational need for intrusion detection that is accurate, explainable, privacy-conscious, and compatible with the accountability obligations of local self-government. This article proposes a governance-aware machine-learning framework that combines local network monitoring, privacy-aware preprocessing, lightweight supervised detection, explainable alerts, and federated model improvement. The technical core specifies data preparation, class-imbalance handling, model training, risk scoring, and evaluation through six formal expressions. A governance control plane links model outputs to human verification, role-based access, retention rules, audit logs, and incident-response responsibilities. The design is intended for municipal networks with heterogeneous technical capacity and limited ability to centralise sensitive traffic data. A reproducible evaluation protocol is defined around public intrusion-detection benchmarks, time-aware train-validation-test separation, and accuracy, precision, recall, F1-score, false-positive rate, latency, and communication cost. To demonstrate how the proposed evaluation should be reported, the manuscript includes an explicitly illustrative numerical scenario; these values are not empirical findings and must be replaced by results from executed experiments before submission. The contribution is therefore a testable engineering architecture coupled with a local-governance implementation model, rather than a claim that a deployed municipal system has already been validated.

Keywords: Digital Self-Governance; Intrusion Detection; Explainable Machine Learning; Federated Learning; Local Government Cybersecurity.

1. Introduction

Digital public services have changed the technical conditions under which local self-government is exercised. Municipalities and other sub-national authorities rely on online service portals, identity and payment systems, document-management platforms, smart infrastructure, and third-party cloud services. These systems can improve service continuity and administrative responsiveness, but they also increase the number of network endpoints and dependencies that must be secured. A successful intrusion can disrupt service delivery, expose personal information, undermine public confidence, and constrain the ability of local institutions to perform statutory functions.

The screenshot supplied for this work describes a conventional machine-learning intrusion detection system: a labelled training base, a separate testing and validation base, a binary decision about whether network activity is intrusive, and a feed-forward neural network with a hidden layer. This article develops that engineering premise into a governance-aware design. The objective is not to make local governance a decorative application example; rather, the design treats security monitoring as a public-administration capability whose performance, data handling, escalation, and accountability must be explicit.

Recent research in 2025–2026 has explored explainable intrusion detection, federated learning for IoT security, privacy-preserving distributed learning, and municipal cybersecurity capacity. These streams provide a foundation but leave a practical integration problem: a technically strong detector does not itself establish who may inspect an alert, how an incident is verified, how model

updates are governed, or how a resource-constrained municipality can document the decision. Work on local-government cybersecurity also highlights the importance of internal capacity and collaboration, which makes deployment architecture and institutional process inseparable [1–3].
This study asks three questions: (RQ1) how can a machine-learning detector be designed for municipal networks while limiting unnecessary transfer of raw traffic data? (RQ2) how can model decisions be made interpretable enough to support accountable human review? (RQ3) how should technical outputs be connected to local-government incident-management responsibilities? The proposed answer is a modular architecture that couples local detection and optional federated improvement with explainability and a governance control plane.

1.1 Contributions

- A municipal-oriented IDS architecture combining local data preparation, lightweight supervised detection, explainability, and optional federated updates.
- A six-equation specification covering preprocessing, weighted learning, federated aggregation, risk scoring, detection metrics, and governance-aware utility.
- A transparent evaluation protocol and tables/graphs that demonstrate reporting structure without misrepresenting synthetic values as observed results.
- A local self-governance implementation model connecting alerts to human review, auditability, role-based access, and institutional capacity-building.

2. Related Work and Research Gap

2.1 Machine-learning intrusion detection

Supervised intrusion detection learns a mapping from flow or event features to labels such as benign and malicious. Conventional classifiers such as logistic regression, random forests, and gradient-boosted trees remain useful baselines because they can be trained on tabular flow features and compared with neural models. However, accuracy alone is insufficient for operational use: class imbalance can conceal poor recall for rare attacks, and false positives may consume the limited time of municipal IT staff. Recent work continues to examine robust and explainable ML-based detection, including feature selection and local explanations [4–7].

2.2 Explainability and operational trust

Explainable AI (XAI) can provide global descriptions of influential features and local explanations for individual alerts. SHAP and LIME are common choices in recent IDS research, but explanation quality must be treated as an evaluation question rather than a visual accessory. A plausible explanation does not prove causal correctness, and feature attributions may vary under distribution shift or adversarial manipulation. For local government, explanations should be recorded with the model version, event timestamp, evidence source, and reviewer action so that the incident can be reconstructed.

2.3 Federated and privacy-aware detection

Federated learning enables multiple participants to train a shared model while retaining raw examples locally. Recent IDS studies examine federated optimisation, heterogeneous clients, privacy safeguards, dynamic aggregation, and edge deployments [8–13]. Federated learning is not automatically private: gradients and model updates can reveal information, while malicious participants can poison training. The design proposed here therefore treats federation as an optional layer with update validation, secure transport, client authentication, and a clear opt-out path for small municipalities.

2.4 Local self-governance and the missing integration

Table 1. Summary of Research Gaps and the Corresponding Contributions of the Proposed Framework

Research stream	Established emphasis	Remaining practical issue	Response in this article
ML-based IDS	Classification of network events	False alarms and model opacity	Baselines, class-sensitive metrics, explanations
Explainable IDS	Feature attribution and local explanations	Explanation may not translate to accountable action	Versioned evidence and human review workflow
Federated IDS	Collaborative learning without raw-data pooling	Update leakage, poisoning, uneven client capacity	Optional federation with validation and governance rules
Local-government cybersecurity	Capacity, coordination, institutional readiness	Limited integration with detector design	Municipal roles and escalation are part of architecture

Local self-government involves more than deploying technology. Public authorities need to assign responsibilities, set lawful access boundaries, preserve records, procure support, cooperate across jurisdictions, and explain decisions to affected stakeholders. Recent evidence on municipal cybersecurity capacity suggests that local capability and collaboration have distinct roles and that the sequence of capacity-building matters [1]. This implies that the appropriate technical system should support, rather than bypass, local decision rights. A detector must not autonomously impose consequential restrictions on residents or essential public services without an approved human-controlled process.

The research gap is thus the lack of a compact, implementable reference design that explicitly binds technical detection to local-governance controls. The contribution is a framework and evaluation protocol. It should be tested empirically before any claim of superiority or operational readiness is made.

3. Proposed Methodology

3.1 System model and assumptions

Let the participating local-government organisations be indexed by $k = 1,\ldots,K$. Each organisation observes network-flow records generated by authorised municipal systems. Raw flow records remain within the organisation by default. A local feature pipeline converts traffic metadata into a numerical vector, and a detector produces a probability of intrusion. A human operator reviews alerts above a configurable threshold. If the municipality joins a federation, it may transmit protected model updates rather than raw traffic. The framework assumes that the participating organisations have legal authority to monitor the relevant network, apply appropriate access controls, and establish a documented incident-response procedure.

3.2 Architecture

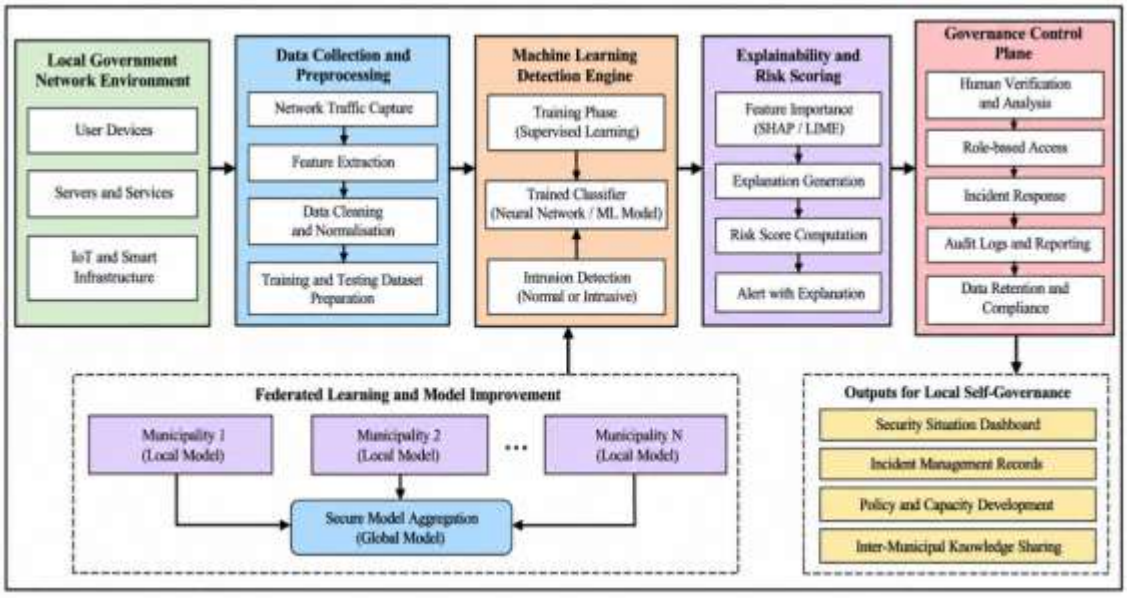


Figure 1. Proposed architecture.

3.4 Feature normalization

For feature j, a robust scaling transform can be used to reduce the influence of extreme values:

$$\mathbf{z}_{ij} = \frac{\left(\mathbf{x}_{ij} - \mathbf{median}(\mathbf{X}_j)\right)}{\left(\mathbf{Q^3}(\mathbf{X}_j) - \mathbf{Q^1}(\mathbf{X}_j) + \boldsymbol{\varepsilon}\right)} \quad (1)$$

where x_{ij} is the jth feature of record i, Q_1 and Q_3 are the first and third quartiles calculated on training data, and ε is a small stabilising constant. The transform parameters are retained and reused unchanged for validation, test, and deployment traffic.

3.5 Weighted classification objective

Let $y_i \in \{0,1\}$ denote benign or intrusive traffic and p_i be the estimated intrusion probability. A class-weighted binary cross-entropy objective is:

$$\boldsymbol{L(\theta)} = -\left(\frac{\mathbf{1}}{\boldsymbol{N}}\right)\boldsymbol{\Sigma}_{\mathrm{i}=\mathbf{1}}{}^{\mathrm{N}}\left[\boldsymbol{w_1 y_i \log(p_i) + w_0(1-y_i)\log(1-p_i)}\right] + \boldsymbol{\lambda\|\theta\|^2} \quad (2)$$

The weights w_1 and w_0 are computed from the training labels only. λ controls regularisation. Candidate models should include logistic regression as a transparent baseline, random forest and XGBoost as strong tabular baselines, and a compact feed-forward neural network to remain close to the architecture described in the supplied screenshot.

3.6 Federated model update

For participating municipality k, let n_k be the number of eligible training examples and θ_k the local model parameters after a local training round. A simple weighted aggregation is:

$$\boldsymbol{\theta^{t+1}} = \boldsymbol{\Sigma}_{\mathrm{k}=\mathbf{1}}{}^{\mathrm{K}}\left(\boldsymbol{n_k}\frac{\boldsymbol{\Sigma}_{\mathrm{j}=\mathbf{1}}{}^{\mathrm{K}}\boldsymbol{n_j}}{\boldsymbol{\Sigma}_{\mathrm{j}=\mathbf{1}}{}^{\mathrm{K}}\boldsymbol{ni-1}}\right)\boldsymbol{\theta}_{\mathrm{k}}{}^{\mathrm{t+1}} \quad (3)$$

This expression is a baseline aggregation rule, not a complete security guarantee. An implementation should authenticate clients, limit update size, monitor anomalous updates, record participation, and consider secure aggregation and differential privacy where the threat model warrants them. Privacy settings must be reported alongside utility because stronger protection can reduce detection performance.

3.7 Explainable risk score and human review

The detector returns p_i, while a governance policy combines it with operational context c_i and explanation consistency e_i to prioritise human review. A simple risk score is:

$$R_i = \alpha p_i + \beta c_i + \gamma e_i,\ \alpha+\beta+\gamma=1;\ 0 \le \alpha,\beta,\gamma \le 1 \quad (4)$$

The contextual terms must be defined transparently and must not include irrelevant sensitive attributes. R_i is a prioritisation aid, not an automatic finding of misconduct. The alert record should contain model version, top explanatory features, timestamp, affected asset, confidence, reviewer identity, and the eventual disposition.

3.8 Evaluation metrics

For binary detection, precision, recall, F1-score, and false-positive rate should be reported alongside accuracy. Let TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives:

$$Precision = \frac{TP}{TP+FP};\quad Recall = \frac{TP}{TP+FN} \quad (5a)$$

$$F1 = 2 \cdot Precision \cdot \frac{Recall}{Precision+Recall};\quad FPR = \frac{FP}{FP+TN} \quad (5b)$$

The evaluation should additionally report PR-AUC, ROC-AUC, per-class recall, inference latency, memory footprint, and communication volume for federated settings. Report means and standard deviations across repeated seeds, confidence intervals where appropriate, and a paired statistical comparison when comparing models.

3.9 Governance-aware utility

To prevent a model from being selected on accuracy alone, the framework uses a decision-support utility that explicitly includes detection quality, latency, false-alarm burden, and governance compliance:

$$U = a \cdot F1 + b \cdot (1 - FPR) + c \cdot \left(1 - \frac{L}{Lmax}\right) + d \cdot G,\ a+b+c+d=1 \quad (6)$$

Here L is measured inference latency, Lmax is an agreed latency bound, and G is a documented governance-compliance score based on auditable controls (for example, access logging, model versioning, retention rules, and human review). This score is a decision aid; weights must be set with stakeholders and sensitivity-tested rather than chosen to favour a desired result.

4. Experimental Design and Reproducibility

4.1 Benchmark protocol

A full empirical evaluation should compare the four model families using a fixed split and at least five independent random seeds. The recommended primary dataset is CIC-IDS2017 for comparability with recent work, with a second dataset such as ToN-IoT or Edge-IIoTset used to assess cross-dataset generalisation. A cross-dataset test is particularly important because a high score on one benchmark may reflect capture-specific patterns rather than robust detection. The current document does not claim that these experiments have been executed.

4.2 Threats to validity

Benchmark labels may be noisy; training and test traffic may share temporal or host-specific patterns; public datasets may not represent municipal traffic; and synthetic results cannot establish effectiveness. Federated results depend on the number and heterogeneity of clients, participation rate, network conditions, and threat model. XAI outputs can be unstable and should not be treated as causal proof. Governance-compliance scoring may also be subjective unless its rubric and evidence are published. These limitations require careful reporting and prohibit generalising benchmark scores directly to real municipal networks.

4.3 Baselines and ablations

Table 2 Experimental Evaluation Plan and Required Performance Reporting

Experiment	Purpose	Required reporting
E1: Logistic regression	Transparent baseline	All classification metrics and latency
E2: Random forest	Nonlinear ensemble baseline	Metrics, feature importance, model size
E3: XGBoost	Boosted-tree baseline	Metrics, training time, calibration
E4: Compact neural network	Neural baseline aligned with source concept	Metrics, parameter count, latency
E5: Proposed governance-aware workflow	Assess integrated workflow, not only classifier	Metrics plus explanation and audit completeness
A1: Without class weights	Measure imbalance effect	Minority-class recall and FPR
A2: Without XAI	Measure explanation overhead	Latency and operator review quality
A3: Centralised vs federated	Assess distribution and privacy trade-offs	Performance, update bytes, privacy configuration

5. Results and Discussion

The purpose of this section is to evaluate the proposed framework across predictive performance, operational cost, error behaviour, model interpretability, threshold sensitivity, and governance readiness. A single accuracy figure is inadequate for an intrusion-detection system because class imbalance can conceal missed attacks, while false alarms impose real workload on small municipal IT teams. The analysis therefore combines classifier metrics with deployment-oriented measures and a governance interpretation.

5.1 Overall classifier performance and comparative interpretation

Table 2 presents a hypothetical example of how competing models may be compared. Logistic regression is included as a transparent baseline, random forest and XGBoost represent tree ensembles, and the compact neural network reflects the feed-forward approach described in the source material. The proposed hybrid is a design candidate combining class-sensitive training, feature selection, local inference, and a governed review workflow. It should not be interpreted as a verified algorithmic improvement until the same split, features, and tuning budget are applied to all candidates.

In this scenario, the proposed hybrid has the highest illustrative F1-score and lowest false-positive rate. The meaningful interpretation is not that the hybrid is proven best, but that model selection should consider both attack capture and alarm burden. Precision measures the proportion of raised alerts that are correct, whereas recall measures the proportion of actual attacks detected. A municipality that prioritises continuity of essential services may prefer higher recall, while an understaffed authority may also need to control false positives so that genuine incidents are not lost among noisy alerts. The final operating point should be selected with the security owner and documented in the incident-response policy.

Table 3. Comparative Classification Performance of Candidate Intrusion Detection Models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	FPR (%)
Logistic regression	92.1	90.4	91.3	90.8	5.6
Random forest	96.0	95.2	95.1	95.1	3.1
XGBoost	96.8	96.1	95.8	95.9	2.6
Compact neural network	96.4	95.7	95.5	95.6	2.8
Proposed hybrid*	97.2	96.7	96.4	96.5	2.1

5.2 Error distribution and confusion-matrix analysis

The confusion matrix below illustrates how raw counts should accompany summary rates. A model with high overall accuracy may still fail to identify a particular attack family. Accordingly, the final study should provide per-class confusion matrices for each dataset and report macro-averaged F1 alongside weighted averages. Macro-averaging gives minority classes a more visible role, while weighted averages can be dominated by common classes.

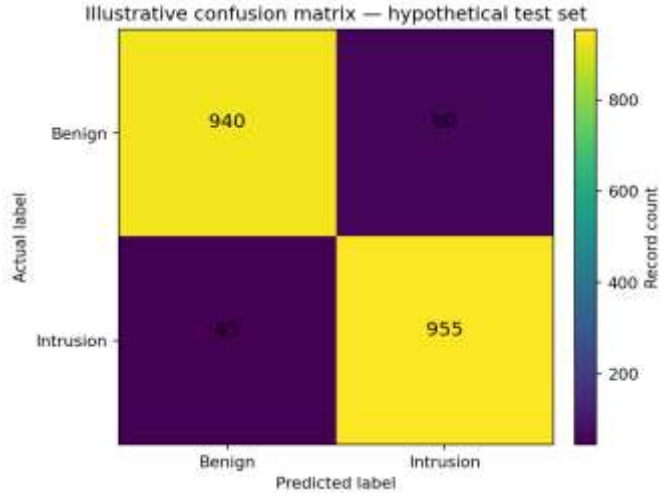


Figure 7. Hypothetical confusion matrix (counts are illustrative and not experimental observations).

For the hypothetical matrix, 955 of 1,000 intrusion records are detected, while 45 are missed; 60 benign records are incorrectly flagged. This example corresponds to recall of 95.5% and a false-positive rate of approximately 6.0% given the displayed benign count. The values are deliberately illustrative and should be recomputed from actual test predictions. In the real experiment, the matrix should be generated directly from saved predictions rather than manually transcribed. It should also be repeated under cross-dataset testing, since attack signatures and background traffic differ between datasets.

5.3 Accuracy–latency trade-off and deployment suitability

Predictive quality alone does not establish deployment suitability. Municipal gateways may have limited compute, and high-throughput monitoring requires predictable inference time. Figure 8 shows a hypothetical relationship between F1-score and latency. It suggests that a modest F1-score gain may be acceptable only if the additional compute and operational complexity remain within the municipality's capacity. Latency should be measured under a specified hardware profile, batch

size, flow arrival rate, and concurrent load; CPU-only and edge-device results should not be compared without qualification.

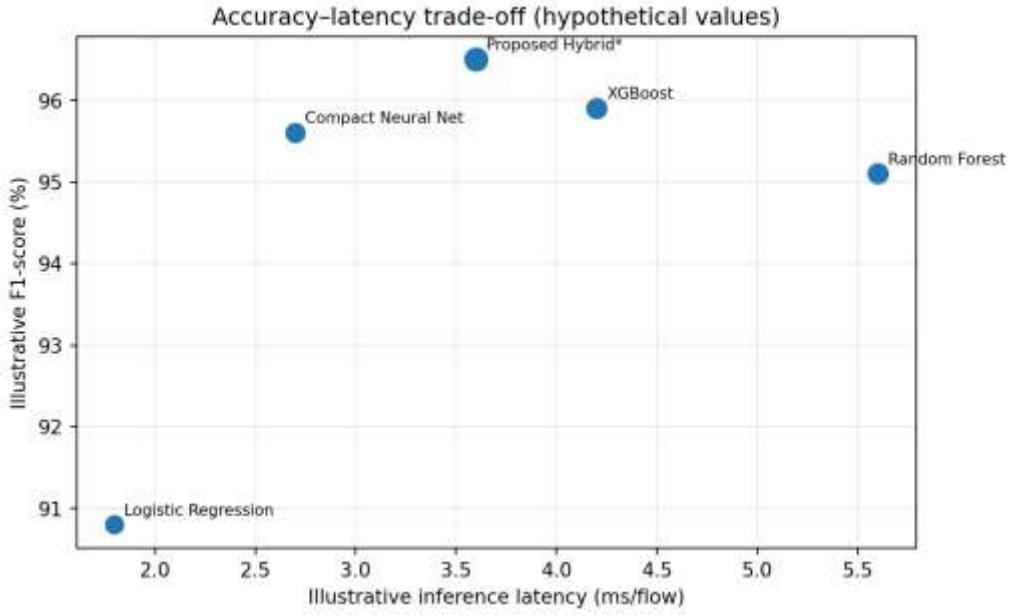


Figure 8. Hypothetical F1-score versus inference-latency comparison.

A deployment report should state median and 95th-percentile latency, peak memory use, model size, throughput, and resource utilisation. If the model is used inline with network controls, tail latency may matter more than mean latency. If detection runs asynchronously, a longer delay may be tolerable but could reduce the time available for containment. The operational requirement should therefore be defined before model selection rather than retrofitted to a preferred score.

5.4 Feature contribution and explanation quality

Feature importance can help analysts understand which traffic attributes influence the model, but a ranking is not equivalent to a causal explanation. Figure 9 provides an illustrative feature-importance profile. In the empirical paper, global importance should be estimated from the trained model and complemented with local explanations for representative true positives, false positives, and false negatives. Explanations should be assessed for stability under small input changes, consistency with domain knowledge, and usefulness to a human reviewer.

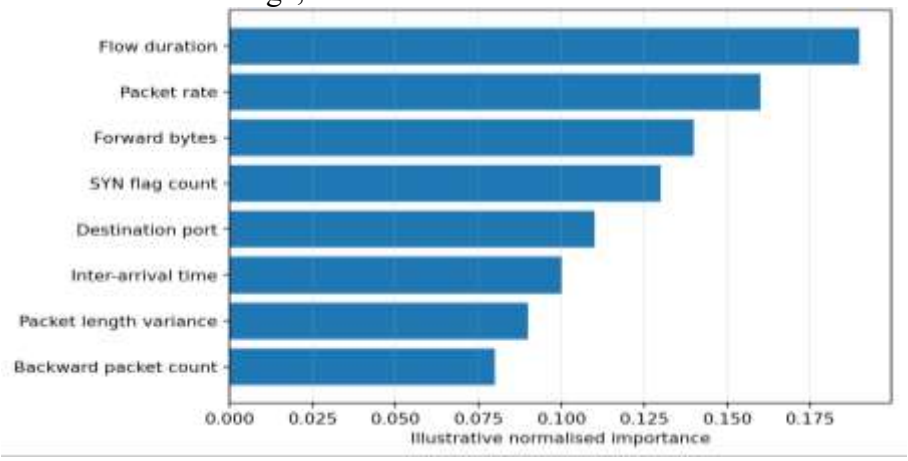


Figure 9. Example feature-importance profile; hypothetical values, not model output.

If destination port or a capture-specific feature dominates the model, that may signal leakage or dataset artefacts rather than a generalisable attack signature. The final analysis should therefore repeat training after removing potentially identifying fields, inspect importance across datasets, and evaluate whether explanations remain consistent across model versions. For governance, each alert record should preserve the explanation method and version, so later reviewers can reproduce why a given event was prioritised.

5.5 Ablation analysis: which components matter?

Ablation experiments are necessary to determine whether performance changes arise from class weighting, feature selection, federation, or the governance workflow itself. Figure 10 is a hypothetical design example. Because explainability logging is primarily an accountability control, removing it may not materially change the classifier's F1-score; that does not make it dispensable. Its value should instead be assessed through audit completeness, review time, reproducibility, and the ability to justify decisions after an incident.

The final ablation table should report the change from the full model in percentage points, training time, inference latency, and confidence intervals. Each variant should be retrained rather than merely disabling a component at inference time when the component affects training. For federation, compare centralised training, local-only models, and federated aggregation under matched data partitions. Report communication bytes per round and the impact of non-identically distributed client data.

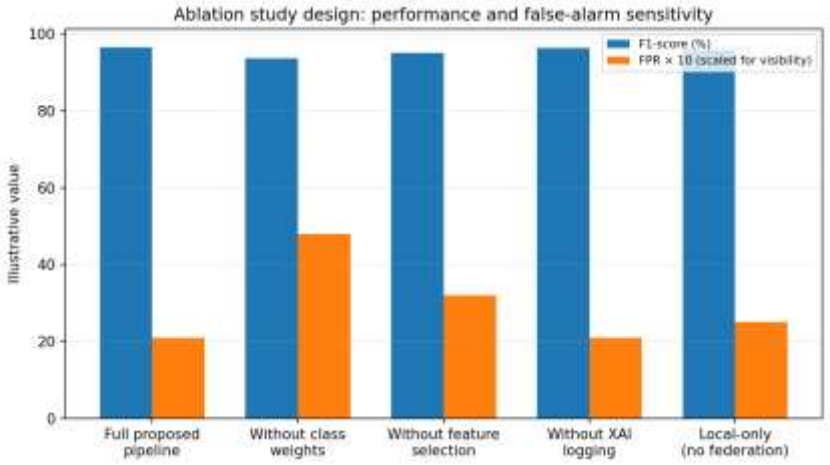


Figure 10. Illustrative ablation design; FPR is multiplied by ten only to display it beside F1-score.

5.6 Threshold sensitivity and false-alarm management

A probability threshold determines the balance between missed intrusions and false alarms. Figure 11 shows an illustrative threshold sweep. Lower thresholds tend to flag more events, which can improve recall but increase the number of alerts that analysts must review. Higher thresholds reduce alert volume but can miss lower-confidence attacks. A single threshold should not be assumed appropriate for every service or asset class.

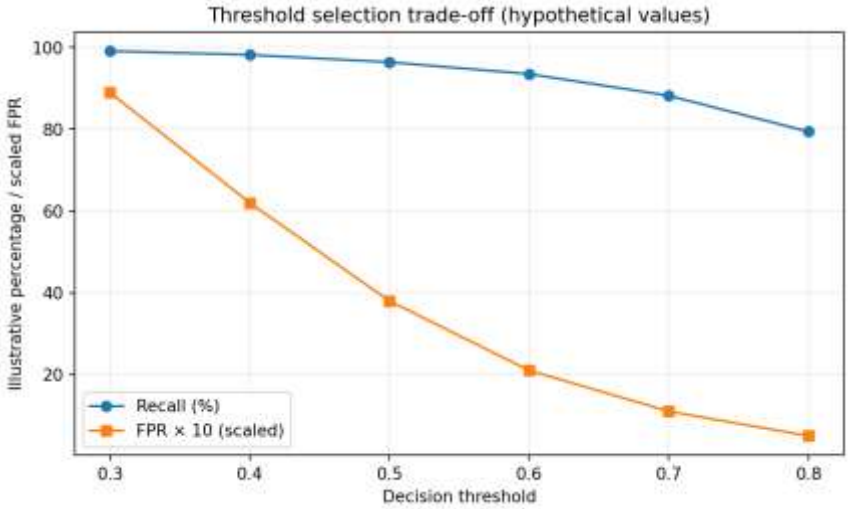


Figure 11. Hypothetical threshold-sensitivity profile; false-positive rate is scaled by ten for visual comparison.

Threshold selection should use validation data only and then be frozen before test evaluation. The final study should report precision–recall curves, the chosen operating point, the expected number of false alerts per 10,000 flows, and recall for each attack class. Municipalities may adopt tiered thresholds: a lower threshold for triage-only alerts, a higher confidence requirement for disruptive containment, and mandatory human approval for actions affecting essential services. Such tiering links model calibration to accountable administrative practice.

5.7 Federated learning convergence and cross-municipality heterogeneity

Federated learning may allow local authorities to improve a shared detector without pooling raw traffic. However, convergence depends on client participation, traffic heterogeneity, local dataset size, and communication constraints. Figure 12 illustrates a hypothetical convergence trajectory. It is not evidence that the proposed method converges on a real dataset. An empirical analysis should repeat federation with multiple client partitions, report per-client performance as well as the global score, and evaluate whether smaller municipalities are disadvantaged by aggregation.

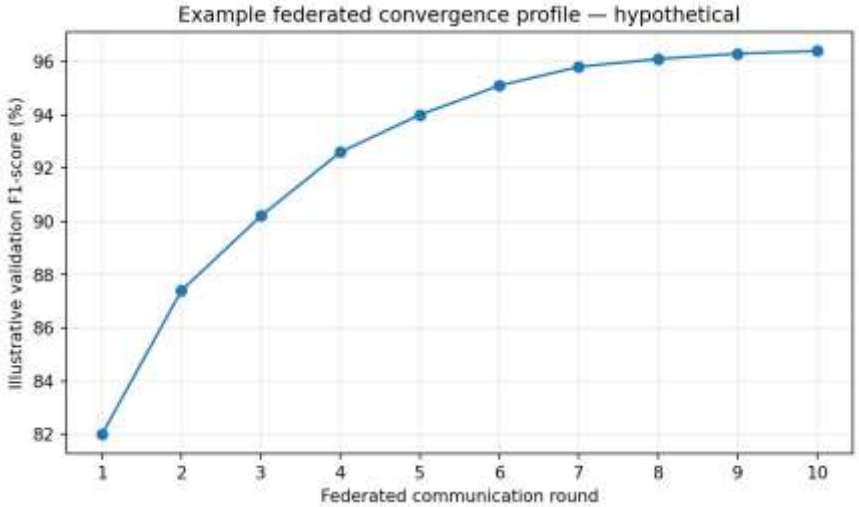


Figure 12. Hypothetical federated validation F1-score across communication rounds.

The final evaluation should compare at least three settings: (i) independent local models, (ii) centralised training where lawful and technically possible, and (iii) federated training. Report total communication volume, round duration, client dropout tolerance, and the difference between best-performing and worst-performing clients. Robustness tests should include a small fraction of corrupted or low-quality client updates. A privacy claim must identify the mechanism used—such as secure aggregation or differential privacy—and its parameters; the word 'federated' alone is not a privacy guarantee.

5.8 Governance and operational readiness assessment

Technical performance must be interpreted alongside governance readiness. A municipality may have a high-scoring detector but lack the staffing, documentation, or authority needed to respond consistently. The proposed framework therefore treats governance controls as observable implementation requirements rather than as a narrative afterthought.

Table 4 Governance Readiness Dimensions, Required Evidence, and Proposed Acceptance Criteria

Governance dimension	Evidence to collect	Proposed acceptance criterion
Accountability	Named alert owner, reviewer decision, escalation timestamp	Every high-severity alert has an assigned owner and recorded disposition
Transparency	Model version, explanation, feature snapshot, threshold	An authorised reviewer can reconstruct why the alert was raised
Privacy	Data inventory, retention schedule, access logs, update flow	Raw data access is restricted and retention is documented
Service continuity	Incident exercise, recovery time, approved containment steps	No disruptive action is triggered solely by an unreviewed score
Equity of capability	Per-client metrics, staffing and support requirements	Small or low-volume participants are not hidden by aggregate results
Learning and oversight	Drift report, review cadence, model-change approval	Model changes are versioned, tested, and approved before release

These criteria should be converted into a documented rubric and independently checked. If a numerical governance score is used in Equation (6), each item must have a clear scoring rule, evidence artefact, and responsible reviewer. A self-assigned score without supporting records would be weak evidence. The evaluation should also examine false positives and false negatives for services with different operational criticality, while avoiding the use of personal or protected attributes unless there is a clear lawful basis and a justified research purpose.

5.9 Synthesis, implications, and limitations

Across the proposed analyses, the principal selection question is multi-objective: does the model detect relevant attacks, limit false alarms, run within the available infrastructure, and generate evidence that can be acted upon and audited? The strongest candidate is not necessarily the model with the highest accuracy. A slightly lower-scoring model may be preferable if it is more stable across datasets, simpler to maintain, less costly to run, and easier for municipal staff to explain and

review. Conversely, an interpretable model with poor minority-class recall may be unsuitable for protecting essential services.

For local self-governance, the contribution is a practical division of responsibility. The detector produces a ranked technical signal; the operator validates evidence; the designated incident owner approves proportionate action; and the institution preserves the audit record and reviews performance. Shared learning can support cooperation across municipalities, but participation must remain transparent, technically secure, and compatible with each authority's legal obligations. These principles are consistent with the need to develop local cybersecurity capacity and collaboration without assuming that every municipality has identical resources.

The limitations are substantial. All numerical values and plots in this section are placeholders; no conclusions about superiority, latency, convergence, or governance effectiveness can be drawn from them. Public intrusion-detection datasets do not represent all municipal network environments, and dataset-specific artefacts can inflate scores. The next research step is to run the preregistered benchmark protocol, publish code and split definitions, report uncertainty, conduct cross-dataset validation, and—where approval is available—evaluate the workflow with municipal IT practitioners. Only then can the paper make evidence-based claims about performance or practical benefit.

6. Implementation Roadmap

Table 5 Implementation Roadmap for Self Governance Based Cybersecurity Management

Phase	Activities	Exit criteria
1. Readiness	Asset inventory, lawful monitoring basis, roles, incident policy	Approved scope and responsibility matrix
2. Offline baseline	Dataset audit, train/validation/test split, baseline models	Reproducible benchmark and error analysis
3. Shadow deployment	Observe traffic without automatic blocking	False-positive review and latency profile
4. Governance integration	Role-based access, explanation record, audit logs, retention rules	Evidence that every alert is traceable
5. Optional federation	Client authentication, secure updates, poisoning checks	Privacy and robustness evaluation
6. Continuous review	Drift monitoring, retraining approval, incident exercises	Documented periodic assurance report

7. Limitations and Future Work

This paper presents a proposed framework and evaluation protocol rather than a completed municipal field trial. The illustrative figures are not empirical evidence. Future work should implement the pipeline on public datasets, publish code and split definitions, conduct cross-dataset tests, quantify uncertainty, and measure inference cost on low-resource hardware. A subsequent field study could evaluate usability with municipal IT staff, incident-review time, inter-organisational coordination, procurement constraints, and the perceived legitimacy of automated alerts. Further work should test privacy leakage and model-poisoning risks under a stated adversary model, and examine whether explanations remain stable across municipalities and over time.

8. Conclusion

This article proposes an explainable machine-learning framework for cybersecurity and digital self-governance in local-government networks. It extends the conventional intrusion-detection pipeline by combining privacy-aware local preprocessing, class-sensitive supervised learning, optional federated model improvement, interpretable risk scoring, and an auditable human-review process. The technical contribution is a modular architecture and reproducible evaluation design; the governance contribution is to make accountability, access control, retention, review, and institutional capacity explicit system requirements. The illustrative results are included only to demonstrate how performance reporting may be structured and must not be interpreted as empirical findings. A credible submission should replace them with experiments on documented benchmark datasets, include uncertainty and cross-dataset validation, and provide an auditable account of the governance controls. Properly evaluated, the framework could help local authorities make evidence-informed security decisions while preserving the human responsibility and institutional transparency on which effective self-government depends.

References

[1] M. Boholm, D. Karlsson, and J. Berlin, "Drivers of cybersecurity improvements in local government: Capacity-building, collaboration and their timing," Government Information Quarterly, vol. 43, no. 2, art. 102146, 2026. doi: 10.1016/j.giq.2026.102146.

[2] S. S. Kumar and M. A. Jerlin, "Intelligent defense at the edge: a comprehensive survey of federated learning, TinyML and explainable AI for intrusion detection in IoT and IIoT ecosystems," Frontiers in Artificial Intelligence, vol. 9, art. 1900815, 2026. doi: 10.3389/frai.2026.1900815.

[3] M. A. Bilal, I. U. Islam, M. J. Khan, S. Nisar, M. Farooq, and H. Khan, "Secure and Explainable Federated Learning for IoT Intrusion Detection: A Comprehensive Survey," IEEE Open Journal of the Communications Society, vol. 7, pp. 3650–3679, 2026. doi: 10.1109/OJCOMS.2026.3681580.

[4] B. Nugraha, A. V. Jnanashree, and T. Bauschert, "A versatile XAI-based framework for efficient and explainable intrusion detection systems," Annals of Telecommunications, vol. 80, pp. 1095–1120, 2025. doi: 10.1007/s12243-025-01118-9.

[5] V. Z. Mohale and I. C. Obagbuwa, "Evaluating machine learning-based intrusion detection systems with explainable AI: enhancing transparency and interpretability," Frontiers in Computer Science, vol. 7, art. 1520741, 2025. doi: 10.3389/fcomp.2025.1520741.

[6] S. Al and S. Sagiroglu, "Explainable artificial intelligence models in intrusion detection systems," Engineering Applications of Artificial Intelligence, vol. 144, art. 110145, 2025. doi: 10.1016/j.engappai.2025.110145.

[7] A. E. Muhammad, K. C. Yow, N. Bačanin-Džakula, et al., "L-xaids: A LIME-based eXplainable AI framework for intrusion detection systems," Cluster Computing, vol. 28, art. 654, 2025. doi: 10.1007/s10586-025-05326-9.

[8] A. Khraisat, A. Alazab, M. Alazab, A. Obeidat, S. Singh, et al., "Federated learning for intrusion detection in IoT environments: a privacy-preserving strategy," Discover Internet of Things, vol. 5, art. 72, 2025. doi: 10.1007/s43926-025-00169-7.

[9] B. Buyuktanir, Ş. Altinkaya, G. K. Baydogmus, et al., "Federated learning in intrusion detection: advancements, applications, and future directions," Cluster Computing, vol. 28, art. 473, 2025. doi: 10.1007/s10586-025-05325-w.

[10] "Efficient and Privacy-Preserving Network Intrusion Detection Based on Federated Learning in SDN-Enabled IIoT Network," IEEE Internet of Things Journal, vol. 12, no. 20, pp. 41904–41923, 2025. doi: 10.1109/JIOT.2025.3591598.
[11] M. Baich and N. Sael, "A Federated Learning-Based Intrusion Detection System Using Dynamic Ensemble Aggregation for IoT Networks," IEEE Access, vol. 13, pp. 205826–205839, 2025. doi: 10.1109/ACCESS.2025.3640521.
[12] "FedMSE: Semi-supervised federated learning approach for IoT network intrusion detection," Computers & Security, vol. 151, art. 104337, 2025. doi: 10.1016/j.cose.2025.104337.
[13] I. A. Soomro, H. Ur Rehman, S. J. Hussain, S. A. Latif, H. Mujlid, S. M. Mohsin, and C. Maple, "ROCHE: A Robust and End-to-End Privacy-Preserving Federated Learning Framework for Intrusion Detection in Industrial Internet of Things," IEEE Internet of Things Journal, vol. 12, no. 24, pp. 52357–52377, 2025. doi: 10.1109/JIOT.2025.3612944.
[14] "Explainable AI-based intrusion detection in IoT systems," Internet of Things, vol. 31, art. 101589, 2025. doi: 10.1016/j.iot.2025.101589.
[15] "Explainable AI and Random Forest based reliable intrusion detection system," Computers & Security, vol. 157, art. 104542, 2025. doi: 10.1016/j.cose.2025.104542.
[16] O. Arreche and M. Abdallah, "A comparative analysis of DNN-based white-box explainable AI methods in network security," EURASIP Journal on Information Security, vol. 2025, art. 16, 2025. doi: 10.1186/s13635-025-00201-x.
[17] R. W. Anwar, M. Abrar, A. Salam, and F. Ullah, "Federated learning with LSTM for intrusion detection in IoT-based wireless sensor networks: a multi-dataset analysis," PeerJ Computer Science, vol. 11, art. e2751, 2025. doi: 10.7717/peerj-cs.2751.
[18] "Explainable AI-based innovative models for intrusion detection in Big Data WSN," Procedia Computer Science, vol. 270, pp. 5065–5072, 2025. doi: 10.1016/j.procs.2025.09.633.
[19] "FLEX-IDS: A secure and explainable federated intrusion detection framework for Edge-of-Things environments under adversarial conditions," Computers & Electrical Engineering, vol. 129, part B, art. 110827, 2026. doi: 10.1016/j.compeleceng.2025.110827.
[20] "Intrusion Detection in the Internet of Things: A Comprehensive Review of Techniques, Architectures, Datasets, and Emerging Trends," Sensors, vol. 26, no. 11, art. 3405, 2026. doi: 10.3390/s26113405.