LEAKAGE-AWARE LATENT-DIMENSION SELECTION FOR INTRUSION DETECTION IN AGRICULTURAL IOT NETWORKS.

Abstract

Agricultural Internet of Things (AG-IoT) intrusion detection requires models that limit representation complexity without weakening the credibility of their evaluation. This study investigates whether a learned intrusion-detection representation can be reduced to a sparse latent subset while retaining a useful clean-detection profile. A compact one-dimensional CNN encoder, referred to as CNN-Lite, learns a 128-dimensional latent representation; IWHO-Lite performs binary, sparsity-aware dimension selection; and a Random Forest provides the terminal classification score. Data-dependent preprocessing is confined to the training scope, while model-development decisions remain isolated from score calibration, operating-threshold selection, and final testing. Exact-vector overlaps are additionally audited before evaluation. Across Farm-Flow, CICIoT2023, and UNSW-NB15 using three independent model seeds, IWHO-Lite retained 63.00, 37.67, and 41.67 latent dimensions, corresponding to compression rates of 50.78%, 70.57%, and 67.45%, respectively. Mean clean-test F1-scores were 0.4724, 0.9953, and 0.7480. Mean batch-inference latency ranged from 0.0973 to 0.1344 ms per sample in the recorded experimental environment. Farm-Flow combined high recall with a substantial false-positive burden, highlighting strong dataset dependence. The results support a leakage-aware performance-parsimony characterization of latent-dimension selection rather than universal predictive superiority or hardware-validated efficiency.

Key words:-
Agricultural IoT, Intrusion Detection, Latent-Dimension Selection, CNN-Lite, IWHO-Lite, Random Forest, Leakage-Aware Evaluation, Feature Selection

……..

INTRODUCTION:-

Agricultural production increasingly relies on interconnected sensing, actuation, communication, and decision-support systems. Irrigation controllers, environmental sensors, machinery, livestock-monitoring devices, and supervisory platforms exchange operational data through agricultural Internet of Things (AG-IoT) networks. This connectivity improves monitoring and automation, but it also creates attack surfaces whose compromise may affect both information flows and physical agricultural processes. Intrusion detection is therefore relevant not only to protecting network assets but also to maintaining the continuity and reliability of data-driven agricultural operations [1].

Edge gateways are natural observation points for AG-IoT traffic because they aggregate communications from heterogeneous field devices before forwarding data to cloud or management services. Deploying intrusion detection close to these traffic sources can support timely monitoring, yet gateway environments may offer more limited processing and storage capacity than centralized platforms [2]. This motivates detection pipelines that avoid unnecessary representational complexity. A compact representation, however, should not be equated automatically with an energy-efficient or hardware-ready system. Dimensionality, serialized model size, latency, throughput, memory use, and energy consumption describe different properties and require separate evidence.

Most dimensionality-reduction strategies for intrusion detection operate directly on raw or engineered network features. This is useful when original variables are interpretable and consistently defined, but security datasets often differ substantially in feature schemas, categorical encodings, scales, and traffic semantics. An alternative is to first learn a nonlinear representation and then search for a smaller subset within that learned space. In this setting, the selection problem concerns **latent dimensions rather than raw traffic variables**. A latent coordinate summarizes information produced by the encoder and should not automatically be interpreted as a physical network feature. The central question is therefore whether a learned representation can be made more parsimonious while retaining a useful intrusion-detection profile.

Reliable assessment of that question requires more than an architecture. Data-dependent preprocessing, representation learning, latent-mask selection, score calibration, threshold selection, duplicate observations, and final testing can all influence reported performance. Intrusion-detection reviews have identified feature selection and evaluation methodology as persistent sources of inconsistency [3]. Security-oriented machine-learning research has similarly documented evaluation pitfalls [4], while leakage studies show how information from evaluation data can influence fitted transformations or model-selection decisions and produce optimistic results [5], [6]. The present study therefore confines fitted preprocessing to the authorized training scope and keeps the model-development block separate from subsequent score calibration, operating-threshold selection, and final testing. Exact feature-vector overlaps are additionally audited. The protocol is described as **leakage-aware** because it controls identifiable contamination pathways without claiming to eliminate every possible form of distributional dependence.

The work also represents a distinct methodological stage from our previously published agricultural intrusion-detection study. That study combined a CNN and dense neural components into a CNN-DNN classifier and used the Improved Wild Horse Optimizer (IWHO) for hyperparameter optimization on Farm-Flow [7]. The present study assigns the components different roles. A compact one-dimensional encoder, referred to as CNN-Lite, learns a 128-dimensional latent representation; IWHO-Lite searches for a binary subset within that representation; and a Random Forest provides terminal classification. The research problem is therefore not hyperparameter optimization of the earlier CNN-DNN classifier, but explicit control and characterization of representation dimensionality before downstream classification.

Existing work covers feature selection, representation learning, metaheuristic optimization, compact classifiers, and evaluation integrity, but their combination leaves a more specific question unresolved. It is not sufficient to show that an input feature set can be shortened or that a compact neural representation can be learned. What remains to be characterized is whether **an already learned intrusion-detection representation can itself be sparsified**, how much of that representation can be removed, what clean detection profile remains, and whether representation reduction translates into corresponding changes in measured inference properties. These questions must also be addressed without allowing final-test information to guide model-development or operating decisions.

Accordingly, this study addresses the following research question:

To what extent can IWHO-Lite reduce a learned CNN-Lite representation to a sparse latent subset while retaining useful clean intrusion-detection performance under leakage-aware evaluation across agricultural IoT and reference network datasets?

The analysis further examines how much of the original 128-dimensional representation can be removed, how selected dimensionality and the resulting performance vary across independent model seeds, and how the selected representation compares descriptively with the corresponding full-latent configuration under the recorded experimental conditions.

The study makes four contributions. First, it defines a modular AG-IoT intrusion-detection design in which CNN-Lite is used specifically for representation learning, IWHO-Lite performs latent-dimension selection, and a Random Forest provides terminal classification. Second, it implements IWHO-Lite as a binary, sparsity-aware search over a learned 128-dimensional latent space, distinct from the hyperparameter-optimization role of IWHO in our earlier work. Third, it provides a three-seed performance-parsimony characterization across Farm-Flow, CICIoT2023, and UNSW-NB15, combining selected dimensionality and compression with clean detection metrics, serialized model size, measured batch latency, and throughput. Fourth, the analysis is supported by a leakage-aware evaluation protocol using training-scoped fitted transformations, exact-vector overlap auditing, and dedicated downstream roles for calibration, operating-threshold selection, and final testing.

The remainder of the paper reviews related work, presents the framework and experimental methodology, reports latent-dimension selection and clean detection results together with measured inference characteristics and contextual comparisons, discusses their implications and limitations, and concludes with directions for future evaluation.

RELATED WORK:-

Intrusion Detection in Agricultural and Edge IoT Environments:-

Smart-agriculture cybersecurity includes device authentication, secure communication, privacy, access control, anomaly detection, and intrusion detection. Agricultural environments combine conventional network threats with cyber-physical dependencies because communication failures or manipulated telemetry may affect sensing, control, and operational decision-making [1]. AG-IoT architectures also distribute computation across perception, edge, and cloud layers, creating several possible locations for monitoring and response.

Edge-oriented IDS research increasingly examines compact or staged models that place part of the detection process near data sources. Kuo and Wu proposed a lightweight two-stage framework for edge-based IoT detection [8], whereas Mazinani et al. evaluated deep-learning intrusion detectors directly on a constrained IoT device [9]. Such work establishes an important boundary for the present study. Architectural compactness and measured software-level inference characteristics are informative, but they are not substitutes for direct measurements of device memory, electrical energy, or deployment timing.

Dataset choice also affects the interpretation of IDS performance. Farm-Flow provides the domain-specific anchor because it was designed for flow-based intrusion detection in smart-agriculture networks [10]. CICIoT2023 supplies a much larger IoT benchmark with broad attack coverage [11], while UNSW-NB15 provides a complementary general network-security distribution [12]. Using these datasets together exposes the pipeline to different feature spaces, traffic structures, and class distributions. It does not imply that performance obtained on a generic IoT or network dataset transfers automatically to an agricultural deployment.

Feature Selection and Learned Representations:-

Feature selection is widely used to remove redundant or weakly informative inputs before intrusion classification. Reviews and recent IoT studies describe filter, wrapper, embedded, and metaheuristic approaches that trade predictive performance against the number of retained variables [3], [13]-[15]. When selection operates on raw variables, retained inputs maintain their original network semantics. The resulting subset, however, remains closely tied to the schema, encoding, and measurement process of the dataset.

Representation learning provides a different route to dimensionality control. An encoder can transform heterogeneous inputs into a nonlinear latent space before classification. Abdel-Rahman et al. combined feature-importance guidance with an autoencoder to derive reduced IDS representations [16], while Karthikeyan et al. combined metaheuristic feature selection with ensemble representation learning for IoT attack detection [17]. These studies reinforce the potential value of learned compact representations.

Learning a lower-dimensional representation and **selecting coordinates from an already learned representation** nevertheless remain distinct operations. In the latter case, the encoder first establishes a latent basis and a second mechanism determines which coordinates are retained for the terminal classifier. This separation enables representation learning and representation sparsification to be analyzed independently.

Latent coordinates also require more cautious interpretation than raw variables. Their organization depends on encoder parameters, and independently trained encoders may distribute information differently across their latent bases. The present study therefore focuses on selected dimensionality, compression, detection behavior, model size, and measured inference characteristics rather than assigning a physical network meaning to individual latent indices.

Metaheuristic Selection and the Position of IWHO-Lite:-

Subset selection is combinatorial, making exhaustive evaluation impractical even for a moderately sized latent representation. Metaheuristic approaches address this problem by exploring candidate subsets through population-based search while evaluating a fitness function that combines predictive quality and parsimony. Recent reviews and IoT studies illustrate the diversity of such strategies [13]-[15], [17]. Their results also show why optimizer comparisons require caution: datasets, feature spaces, classifier budgets, stopping rules, and validation procedures often differ.

The Wild Horse Optimizer (WHO) was introduced as a population-based metaheuristic for engineering optimization [18]. The Improved Wild Horse Optimizer subsequently introduced modifications intended to strengthen exploration and exploitation [19]. Both provide the optimization lineage from which the present lightweight binary selection mechanism is derived.

IWHO-Lite is used here for a narrower purpose than general continuous optimization. It operates over a binary mask associated with 128 learned CNN-Lite coordinates and evaluates that mask using a predictive-plus-sparsity objective on the model-selection role. The study does not attempt to establish IWHO-Lite as universally superior to WHO, IWHO, or other metaheuristics. Its usefulness is evaluated through the latent subsets it produces and the detection and inference profiles associated with those subsets.

Evaluation Integrity and Decision Control:-

Multi-stage IDS pipelines introduce several possible routes for information leakage. Preprocessing, feature or representation selection, duplicate observations, hyperparameter tuning, and validation reuse can all contaminate an estimate when evaluation information influences model construction [4]-[6]. This issue becomes especially relevant when representation learning, subset search, score transformation, threshold choice, and final evaluation are combined in a single pipeline.

The protocol used here confines fitted preprocessing to the training scope and explicitly separates model development from subsequent calibration, threshold selection, and final testing. The **model-selection role supports model-development decisions**, including CNN-Lite training control and IWHO-Lite mask evaluation. Calibration and threshold selection then use dedicated downstream roles, and the final test population remains untouched until the complete pipeline has been frozen.

Classifier scoring and operating-point selection introduce additional, but secondary, decisions. Random Forest vote proportions are not guaranteed to be calibrated probabilities. Logistic score mapping provides one way of fitting a monotonic relationship between model score and observed outcome [20]. Its usefulness depends on the data and learning algorithm [21], and calibration errors have also been documented in modern neural classifiers [22]. Calibration should therefore be measured rather than presumed.

In the present method, logistic mapping is fitted on the calibration role, after which a fixed F1-oriented threshold is selected on the threshold role. Neither stage is treated as an independent methodological contribution. In particular, the threshold is not dynamic, online, or cost-sensitive.

Research Gap and Positioning:-

The literature therefore provides strong foundations for AG-IoT intrusion detection, compact edge-oriented models, raw-feature selection, representation learning, metaheuristic subset search, and leakage-aware machine-learning evaluation. What remains less directly characterized is their intersection: **explicit search for a sparse subset within a learned intrusion-detection representation, followed by joint characterization of representation size, clean detection behavior, and measured inference properties under controlled data roles**.

The unresolved question is not whether another optimizer can simply be attached to another classifier. It is whether latent-space selection can remove a substantial proportion of a learned representation, what clean detection profile remains after that reduction, and how the resulting detection-complexity relationship varies across distinct traffic distributions. The CNN-Lite-IWHO-Lite-Random Forest pipeline developed here is evaluated specifically for that purpose.

MATERIALS AND METHODS:-

Proposed Framework and Intended AG-IoT Setting:-

The intended deployment context contains three logical layers consistent with common AG-IoT architectures [23]. A perception layer contains sensors, actuators, machinery, and local communication devices. An edge-gateway layer aggregates network flows and represents the intended observation point for the frozen intrusion-detection pipeline. A cloud and management layer can support model training, storage, periodic updates, and broader security analytics.

The gateway is an intended architectural location rather than a hardware platform validated in this study. The experiments characterize latent dimensionality, serialized model size, batch latency, and throughput in the recorded computational environment; they do not establish electrical-energy consumption, peak-memory requirements, thermal behavior, battery autonomy, or real-time performance on an agricultural gateway.

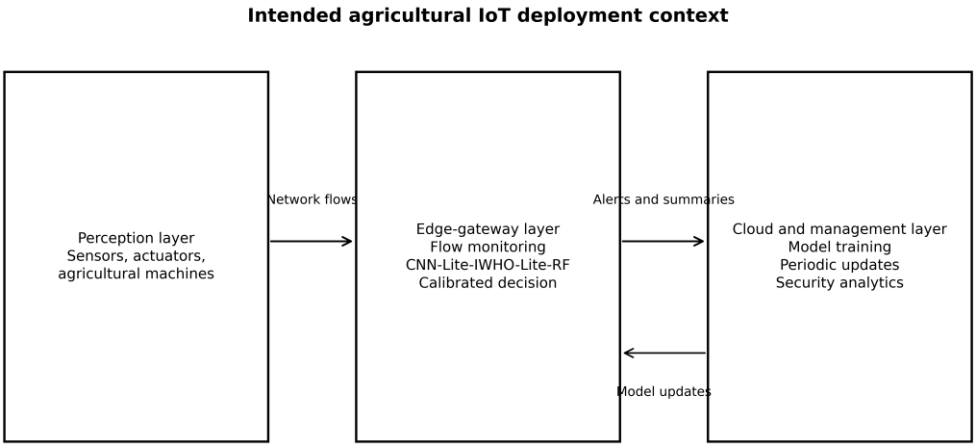

Fig. 1. Intended agricultural IoT deployment context. The perception layer supplies networked sensing and actuation functions. Flow-level monitoring and the frozen intrusion-detection pipeline are positioned at an intended edge gateway, while model training and periodic updates remain cloud-assisted. The diagram describes the target architecture and does not represent a hardware deployment benchmark.

The complete inference sequence is:**training-fitted preprocessing → CNN-Lite → 128-D latent representation → IWHO-Lite selection → k-D selected representation → Random Forest → logistic score mapping → fixed operating threshold.**

Problem Formulation and Notation:-

Let x_i denote the raw feature vector for observation i, with binary target $y_i \in \{0,1\}$. If S_{tr} denotes the authorized training scope, all fitted preprocessing is represented by

$$x_i' = T_{S_{\text{tr}}}(x_i). \tag{1}$$

The CNN-Lite encoder g_θ maps the transformed observation to a 128-dimensional latent vector:

$$z_i = g_\theta(x_i') \in \mathbb{R}^{128}. \tag{2}$$

IWHO-Lite then searches within this representation rather than among the original network variables. The objective is not dimensionality reduction in isolation; latent parsimony and clean detection are examined jointly, with serialized model size and measured inference behavior reported separately.

Leakage-Aware Data Preparation and Experimental Roles:-

The experimental design distinguishes five functions: training, model selection, calibration, threshold selection, and final testing. The model-selection role belongs to the model-development block and can support CNN-Lite training control as well as IWHO-Lite mask evaluation. Calibration and threshold selection use separate downstream roles. The final test population is not used to fit preprocessing, train CNN-Lite, select the latent mask, fit the calibrator, or select the operating threshold.

Numeric variables are imputed using training medians and standardized using means and standard deviations estimated within the training scope. For UNSW-NB15, proto, service, and state are treated as categorical variables, imputed from training-mode values, and one-hot encoded with previously unseen categories ignored. id and attack_cat are removed from

UNSW-NB15, while traffic is excluded from Farm-Flow. Constant variables are identified from the training scope before the fitted transformation is applied unchanged to downstream roles.

Same-label duplicate feature vectors are reduced within fitting data. Exact feature-vector overlaps across fitting and evaluation roles are audited and removed. Feature-identical groups associated with conflicting labels are excluded from fitting roles because deterministic supervised learning cannot assign two targets to the same input. Contradictory-label observations belonging to the final test population are retained without relabeling so that their ambiguity remains represented in final performance.

These controls justify the term **leakage-aware** while explicitly acknowledging that exact-vector auditing does not capture every possible temporal, device-level, farm-level, session-level, or distributional dependency.

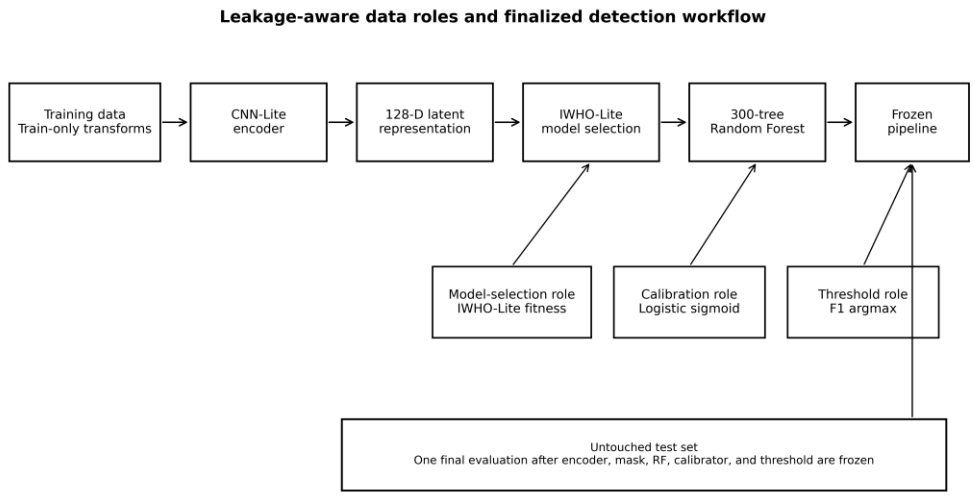

Fig. 2. Leakage-aware data roles and final detection workflow. Fitted preprocessing is confined to the training scope. The model-selection role also supports CNN-Lite training control and early stopping; the arrow to IWHO-Lite highlights its use for mask fitness. Logistic mapping and fixed-threshold selection use dedicated downstream validation roles. Final testing occurs only after the encoder, latent mask, Random Forest, calibrator, and threshold have been frozen.

CNN-Lite Representation Learning:-

CNN-Lite treats each preprocessed flow vector as a one-dimensional feature-axis sequence. The first convolutional block applies 32 one-dimensional filters with kernel size 5, followed by Batch Normalization and ReLU. The second block applies 64 filters with kernel size 3, again followed by Batch Normalization and ReLU. Global Average Pooling summarizes the resulting feature-axis activations, and a dense layer produces 128 latent values followed by Layer Normalization.

The resulting representation is z_i in Equation (2).

During encoder learning, the latent representation is connected to a temporary supervised head containing Dense(128, ReLU), Dropout(0.1), and a sigmoid output. This head is used to train and monitor the encoder; it is not the terminal classifier in the final pipeline.

Training uses Adam with learning rate 10^{-3}, binary cross-entropy, balanced class weights, a maximum of 15 epochs, and batch size 2048. Early stopping and learning-rate reduction are controlled by model-selection performance. The complete CNN-Lite network contains **31,905 total parameters**.

IWHO-Lite Latent-Dimension Selection:-

After representation learning, each candidate latent subset is encoded by a binary mask

$$M \in \{0,1\}^{128}, \qquad k = \| M \|_0, \tag{3}$$

where $M_j = 1$ indicates that latent coordinate j is retained.

The actual selected representation supplied to the downstream classifier is

$$z_i^{(M)} = \left(z_{ij} : M_j = 1\right) \in \mathbb{R}^k. \tag{4}$$

Equation (4) uses explicit coordinate extraction rather than merely setting nonselected entries to zero. This reflects the implemented downstream classifier, which receives only the selected columns.

IWHO-Lite Objective and Search Procedure:-

The IWHO-Lite objective combines model-selection F1 and representation sparsity:

$$J(M) = 1 - F1_{\mathrm{MS}}(M; 0.5) + \lambda_s \frac{k}{128}, \qquad \lambda_s = 0.05. \tag{5}$$

Lower objective values are preferred. The threshold 0.5 in Equation (5) is used only inside mask evaluation and is distinct from the final operating threshold selected later.

The canonical implementation uses a population of 16 candidates, at most 25 iterations per restart, three restarts, and early-stop patience of five iterations with a minimum improvement of 10^{-4}. Candidate masks are sampled using a Bernoulli transfer based on the sigmoid of continuous search positions. Empty masks are forced to retain at least one coordinate.

Candidate fitness uses a successive multi-fidelity Random Forest schedule. At the SCOPUS_READY setting, Level 1 evaluates candidates using up to 10,000 fitting observations and 5,000 model-selection observations with 25 trees and maximum depth 12. The best 25% of candidates are promoted, with at least two retained. Level 2 uses up to 50,000 fitting observations and 20,000 model-selection observations with 50 trees and maximum depth 16, followed by the same 25% promotion rule. Level 3 evaluates the remaining candidates using the configured search limits with 100 trees and unrestricted depth. The same objective in Equation (5) is used at every fidelity.

Algorithm 1. IWHO-Lite binary latent-dimension selection with successive multi-fidelity evaluation:-

Input: training latent matrix Z_{tr}, labels y_{tr}; model-selection latent matrix Z_{MS}, labels y_{MS}; latent dimension $m = 128$; population $P = 16$; maximum iterations $T = 25$; restarts $R = 3$; $\lambda_s = 0.05$.

Output: global-best binary mask M^*.

1. Set global-best objective $J^* \leftarrow +\infty$.
2. For each restart $r = 1, \ldots, R$, initialize positions $X^{(0)} \sim \mathcal{N}(0,1)$ and velocities $V^{(0)} = 0$.
3. For $t = 0, \ldots, T-1$, sample each mask coordinate as $M_{hj} \sim \mathrm{Bernoulli}\left(\sigma\left(X_{hj}^{(t)}\right)\right)$ and enforce a nonempty mask when required.
4. Evaluate all P masks at Level 1 using RF-25 and Equation (5).
5. Rank by $J(M)$ and promote the best 25%, with at least two candidates.
6. Evaluate promoted masks at Level 2 using RF-50; rank and again promote the best 25%, with at least two candidates.
7. Evaluate the remaining masks at Level 3 using RF-100 and identify the iteration-best candidate.
8. If its objective improves the global best, update M^* and J^*.
9. Map the global-best mask to the continuous leader representation $L = 2M^* - 1$.
10. Compute the iteration-dependent inertia $w_t = 0.9 - 0.5\, t/\max(T-1,1)$.
11. Draw $A \sim U(0.2,0.8)$ and $\varepsilon \sim \mathcal{N}(0,0.25)$, then update $V^{(t+1)} = w_t V^{(t)} + A \odot \left(L - X^{(t)}\right) + \varepsilon$ and $X^{(t+1)} = \mathrm{clip}\left(X^{(t)} + V^{(t+1)}, -8, 8\right)$.
12. Stop the restart when no improvement of at least 10^{-4} is obtained for five consecutive iterations, or when $T = 25$ is reached.
13. Return the global-best mask M^* after all restarts.

The test population is not accessed during candidate generation, fidelity evaluation, restart comparison, or final mask selection.

Random Forest Terminal Classification:-

After mask selection, a final Random Forest is fitted on $z_i^{(M)}$. The classifier uses 300 trees, Gini impurity, square-root feature subsampling, bootstrap sampling, unrestricted tree depth, and class weights derived from the complete training distribution. Random Forests are used as nonparametric ensembles of decision trees following the standard formulation of Breiman [24].

The uncalibrated malicious-class score is

$$s_i = h_\phi\left(z_i^{(M)}\right). \tag{6}$$

Latent dimensionality and Random Forest size are analyzed separately because a smaller k does not mathematically imply fewer nodes or a smaller serialized ensemble.

Logistic Score Calibration:-

The Random Forest output s_i is treated as an uncalibrated score. A logistic sigmoid mapping is fitted exclusively on the calibration role:

$$p_i = \frac{1}{1 + \exp[-(\beta_0 + \beta_1 s_i)]}. \tag{7}$$

Calibration is retained as a separate procedural stage rather than an assumed improvement. Its out-of-fit behavior is evaluated using Brier score and Expected Calibration Error.

Fixed Operating-Threshold Selection:-

After calibration, a fixed threshold is selected on the dedicated threshold role. Let $\mathcal{T} = \{0.001, 0.002, \ldots, 0.999\}$. The selected operating point is

$$t^* = \arg\max_{t \in \mathcal{T}} F1_{\text{thr}}(t). \tag{8}$$

Final predictions are then

$$\hat{y}_i = \mathbf{1}[p_i \geq t^*]. \tag{9}$$

The calibrator and t^* remain frozen during final-test evaluation. The threshold is fixed; it is not updated online and does not encode an explicit asymmetric operational-cost function.

Latent Compression and Measured Complexity:-

Latent compression is defined as

$$\rho = 1 - \frac{k}{128}. \tag{10}$$

The corresponding compression percentage is $100\rho\%$.

Representation parsimony is reported separately from measured computational properties. The evaluation therefore includes selected dimension count, compression rate, serialized CNN-Lite size, serialized Random Forest size, mean batch latency per sample, and throughput. No composite resource-efficiency score is constructed.

Datasets and Binary-Label Contracts:-

Farm-Flow is the domain-specific dataset in the evaluation because it was developed for network-flow intrusion detection in smart agriculture [10]. Its binary is_attack variable is used as the target, and traffic is excluded from predictors.

CICIoT2023 is a large IoT benchmark containing benign traffic and multiple attack categories [11]. Recognized benign observations are mapped to class 0 and attack observations to class 1.

UNSW-NB15 provides a complementary general network-security distribution [12]. Its binary label variable is used as the target. id and attack_cat are excluded, while proto, service, and state are handled categorically.

The three datasets expose the architecture to different traffic and class structures. They are not treated as proof of universal cross-domain transfer.

Partitions, Seeds, and Ambiguity Audit:-

The experiments use model seeds **42, 2027, and 314159**. Partitioning precedes fitted preprocessing. For each dataset and seed, the validation population is subdivided into 50% model selection, 25% calibration, and 25% threshold selection. The final test population remains outside all model-development and operating-point decisions.

Table 1. Dataset, partition, validation-role, and ambiguity contracts

Dataset	Training N	Validation N	Test N	MS / Calibration / Threshold	Post-cleaning exact overlap	Contradictory-label test vectors
Farm-Flow	3,510	620	2,348	50% / 25% / 25%	0	51 rows in 3 groups
CICIoT2023	498,836-498,924	149,183-149,272	1,163,577-1,163,586	50% / 25% / 25%	0	0
UNSW-NB15	85,689	15,122	52,746	50% / 25% / 25%	0	8 rows in 4 groups

The ambiguity audit yields a lower bound of 18 irreducible errors among the three contradictory Farm-Flow groups and four among the four UNSW-NB15 groups. These observations are retained in the final test population rather than relabeled.

Training and Optimization Configuration:-

Experiments were executed in a Google Colab workflow with deterministic global seeds and recording of the software and device environment. TensorFlow supported CNN-Lite representation learning, whereas scikit-learn supported preprocessing, Random Forest classification, and logistic score mapping. Latency measurements characterize the recorded execution environment rather than a device-independent guarantee.

Table 2. Final pipeline configuration and data roles

Component	Final configuration	Role
Representation mode	Feature-axis one-dimensional convolution	Input organization
CNN-Lite block 1	Conv1D(32, kernel=5) → BatchNorm → ReLU	Training scope; model-selection monitoring
CNN-Lite block 2	Conv1D(64, kernel=3) → BatchNorm → ReLU	Training scope; model-selection monitoring

Component	Final configuration	Role
Latent layer	GlobalAveragePooling1D → Dense(128) → LayerNorm	128-D representation
Temporary CNN head	Dense(128, ReLU) → Dropout(0.1) → Dense(1, sigmoid)	Encoder training only
CNN optimization	Adam ; BCE; balanced class weights; max 15 epochs; batch 2048	Model development
CNN-Lite size	31,905 total parameters	Fixed architecture
IWHO-Lite objective		Model-selection role
IWHO-Lite search	Population 16; max 25 iterations; 3 restarts; patience 5; minimum improvement	Model-selection role
Multi-fidelity mask evaluation	25 → 50 → 100 trees; successive promotion	IWHO-Lite fitness only
Final Random Forest	300 trees; Gini; max_features=sqrt; bootstrap; unrestricted depth	Selected latent space
Score calibration	Logistic sigmoid mapping	Calibration role
Threshold selection	999 candidates, 0.001-0.999; maximize F1	Threshold role
Experimental seeds	42, 2027, 314159	Independent model runs

Evaluation Metrics and Statistical Reporting:-

Clean classification is evaluated through F1-score, precision, recall, specificity, Matthews correlation coefficient, PR-AUC, and balanced accuracy. The combined metric set is necessary because the three datasets differ substantially in class composition. F1 summarizes positive-class precision and recall, specificity characterizes recognition of benign observations, and MCC provides a correlation-based measure useful under imbalance.

Calibration is evaluated using Brier score and Expected Calibration Error. Threshold-role F1 is reported separately from final-test F1 because the two values refer to different populations and serve different purposes.

Representation complexity is summarized by k and ρ. Software-level inference characteristics are summarized through serialized model size, mean batch latency, and throughput. Results are reported as **mean ± sample standard deviation across three independent model seeds**. The three runs provide descriptive inter-run variation and are not used to support population-level hypothesis-test claims.

Contextual Reference Systems and Comparability Restrictions:-

The primary contextual comparison contrasts the **selected-latent Random Forest** with a Random Forest using the complete 128-dimensional CNN-Lite representation.

Raw-feature Random Forest and CNN-Lite end-to-end classification are retained only as secondary reference systems. They are not controlled ablations. The primary selected-latent pipeline uses a 300-tree final Random Forest, whereas the recorded Random Forest reference systems use 150 trees. CICIoT2023 references also use capped 100,000-row training and test cohorts.

Reference differences are therefore interpreted descriptively rather than causally. They cannot isolate the effect of IWHO-Lite or support a superiority claim.

RESULTS:-

Latent-Dimension Selection:-

IWHO-Lite selected substantially smaller representations than the original 128-dimensional CNN-Lite latent space on all three datasets.

Farm-Flow retained **63.00 ± 3.46 dimensions**, corresponding to **50.78% ± 2.71% compression**. CICIoT2023 retained **37.67 ± 1.15 dimensions**, corresponding to **70.57% ± 0.90% compression**. UNSW-NB15 retained **41.67 ± 2.08 dimensions**, corresponding to **67.45% ± 1.63% compression**.

Selected dimension counts varied from 59 to 65 on Farm-Flow, 37 to 39 on CICIoT2023, and 40 to 44 on UNSW-NB15. The corresponding model-selection F1 values were 0.85711 ± 0.00525, 0.99417 ± 0.00018, and 0.90778 ± 0.00231. Best objective values were 0.16750 ± 0.00413, 0.02055 ± 0.00028, and 0.10850 ± 0.00269.

Search effort also varied. The mean number of evaluated masks was 499 ± 162 for Farm-Flow, 565 ± 71 for CICIoT2023, and 477 ± 111 for UNSW-NB15. These counts describe the realized early-stopping and multi-fidelity search behavior rather than intrinsic dataset difficulty.

Table 3. IWHO-Lite latent-dimension selection across three model seeds

Dataset	Original m	Selected k	Range	Compression	MS F1	Best objective	Mask evaluations
Farm-Flow	128	63.00 ± 3.46	59-65	50.78% ± 2.71%	0.85711 ± 0.00525	0.16750 ± 0.00413	499 ± 162
CICIoT2023	128	37.67 ± 1.15	37-39	70.57% ± 0.90%	0.99417 ± 0.00018	0.02055 ± 0.00028	565 ± 71
UNSW-NB15	128	41.67 ± 2.08	40-44	67.45% ± 1.63%	0.90778 ± 0.00231	0.10850 ± 0.00269	477 ± 111

The consistency claim is deliberately limited to **similarly sized subsets within each dataset**. Individual latent indices are not treated as stable semantic features because each independently trained encoder defines its own latent basis.

Clean Detection Performance:-

Farm-Flow produced the most difficult clean operating profile. Recall remained high at **0.9523 ± 0.0189**, but precision was **0.3143 ± 0.0137** and specificity **0.3167 ± 0.0560**. Mean F1 was therefore **0.4724 ± 0.0131**. MCC reached 0.2684 ± 0.0209, PR-AUC 0.65566 ± 0.02007, and balanced accuracy 0.6345 ± 0.0187. The detector consequently identified most attacks while producing a substantial false-positive burden.

CICIoT2023 yielded the strongest positive-class metrics, with **F1 = 0.9953 ± 0.0001**, precision 0.9955 ± 0.0010, recall 0.9951 ± 0.0008, and PR-AUC 0.99994 ± 0.00000. Specificity was lower at 0.8153 ± 0.0411, giving MCC 0.8035 ± 0.0115 and balanced accuracy 0.9052 ± 0.0202.

UNSW-NB15 produced an intermediate profile, with **F1 = 0.7480 ± 0.0183**, precision 0.6043 ± 0.0259, recall 0.9825 ± 0.0081, specificity 0.6369 ± 0.0428, MCC 0.6041 ± 0.0289, PR-AUC 0.86851 ± 0.01677, and balanced accuracy 0.8097 ± 0.0183.

Table 4. Primary clean-test performance

Dataset	F1	Precision	Recall	Specificity	MCC	PR-AUC	Balanced accuracy
Farm-Flow	0.4724 ± 0.0131	0.3143 ± 0.0137	0.9523 ± 0.0189	0.3167 ± 0.0560	0.2684 ± 0.0209	0.65566 ± 0.02007	0.6345 ± 0.0187
CICIoT2023	0.9953 ± 0.0001	0.9955 ± 0.0010	0.9951 ± 0.0008	0.8153 ± 0.0411	0.8035 ± 0.0115	0.99994 ± 0.00000	0.9052 ± 0.0202
UNSW-NB15	0.7480 ± 0.0183	0.6043 ± 0.0259	0.9825 ± 0.0081	0.6369 ± 0.0428	0.6041 ± 0.0289	0.86851 ± 0.01677	0.8097 ± 0.0183

Joint Performance-Parsimony Profile:-

Selected dimensionality and predictive performance do not follow a monotonic relationship. Farm-Flow retained the largest mean subset-63.00 dimensions-yet produced the lowest F1. CICIoT2023 retained only 37.67 dimensions, corresponding to the largest compression rate, while achieving the highest clean F1. UNSW-NB15 retained 41.67 dimensions and produced an intermediate clean profile.

The evidence therefore supports a narrower interpretation: **IWHO-Lite repeatedly produced sparse latent subsets, while the predictive value associated with those subsets remained strongly dataset-dependent**.

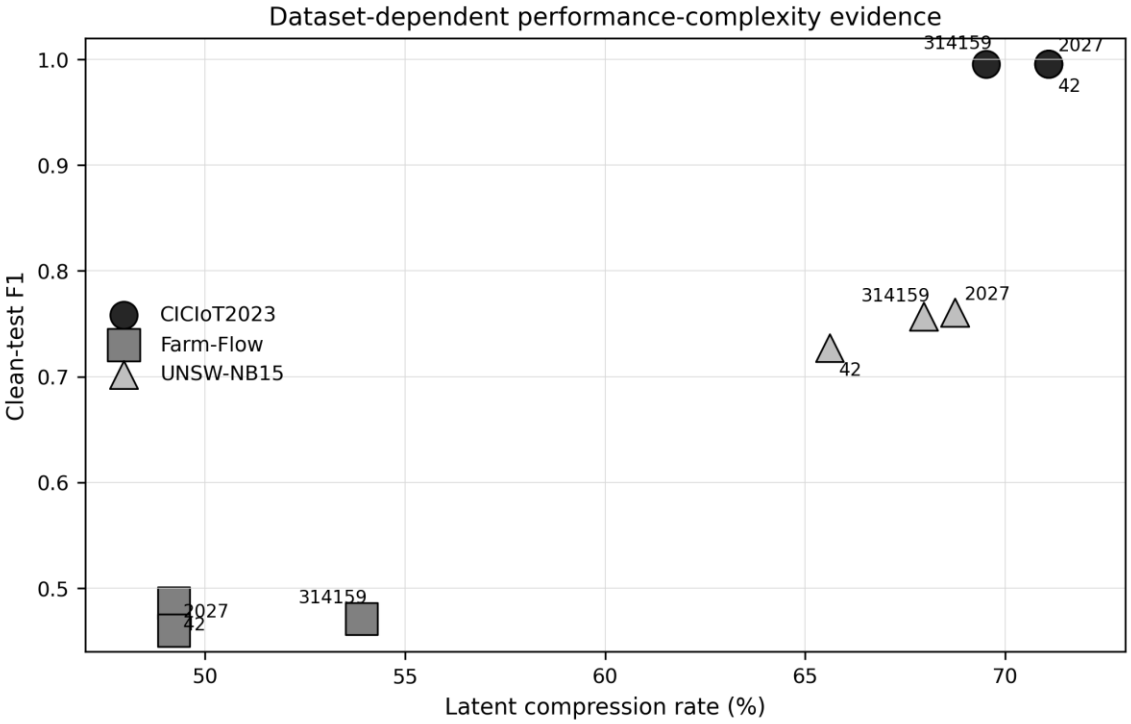

Fig. 3. Clean-test F1 versus latent compression. Each point represents an individual dataset-seed run; marker area reflects measured batch latency. The plot is descriptive and does not define a Pareto frontier. Marker size should not be interpreted as an energy or hardware-efficiency measurement.

Measured Inference Characteristics:-

CNN-Lite contains 31,905 total parameters and occupies approximately **0.097 MiB** in serialized form. The downstream Random Forest size varies substantially across datasets.

Farm-Flow produced a forest of **5.59 ± 0.05 MiB**, CICIoT2023 **39.57 ± 1.07 MiB**, and UNSW-NB15 **57.07 ± 4.22 MiB**. Thus, fewer selected latent dimensions did not imply a smaller terminal classifier. Farm-Flow retained more dimensions than the other two datasets but generated the smallest Random Forest.

Mean batch latency was **0.1344 ± 0.0010 ms/sample** for Farm-Flow, **0.0973 ± 0.0008 ms/sample** for CICIoT2023, and **0.1027 ± 0.0017 ms/sample** for UNSW-NB15. Corresponding throughput was **7,440 ± 57**, **10,282 ± 83**, and **9,743 ± 160 samples/s**.

Farm-Flow latency was measured on its complete 2,348-observation test population. Fixed 5,000-observation cohorts were used for CICIoT2023 and UNSW-NB15.

Table 5. Measured representation and inference characteristics

Dataset	Selected k	Compression	Encoder MiB	RF MiB	Latency ms/sample	Throughput samples/s	Latency cohort
Farm-Flow	63.00 ± 3.46	50.78% ± 2.71%	0.097 ± 0.000	5.59 ± 0.05	0.1344 ± 0.0010	7,440 ± 57	2,348
CICIoT2023	37.67 ± 1.15	70.57% ± 0.90%	0.097 ± 0.000	39.57 ± 1.07	0.0973 ± 0.0008	10,282 ± 83	5,000
UNSW-NB15	41.67 ± 2.08	67.45% ± 1.63%	0.097 ± 0.000	57.07 ± 4.22	0.1027 ± 0.0017	9,743 ± 160	5,000

These measurements characterize the recorded software environment. They do not establish electrical-energy efficiency, peak-memory suitability, or gateway-level real-time guarantees.

Calibration and Fixed Operating Thresholds:-

Calibration is reported as a secondary procedural result.

Mean operating thresholds were **0.604 ± 0.026** for Farm-Flow, **0.638 ± 0.100** for CICIoT2023, and **0.336 ± 0.094** for UNSW-NB15. Their respective ranges were 0.583-0.633, 0.534-0.734, and 0.231-0.412. Threshold-role F1 was 0.88589 ± 0.00408, 0.99576 ± 0.00014, and 0.91736 ± 0.00288.

Logistic mapping did not improve the aggregate out-of-fit Brier or ECE values on any of the three datasets. Farm-Flow Brier score changed from 0.14307 ± 0.00844 to 0.15452 ± 0.00594, while ECE changed from 0.08739 ± 0.01576 to 0.09902 ± 0.01178. CICIoT2023 Brier changed from 0.00587 ± 0.00012 to 0.00646 ± 0.00011 and ECE from 0.00181 ± 0.00024 to 0.00412 ± 0.00036. UNSW-NB15 Brier changed from 0.06046 ± 0.00195 to 0.06155 ± 0.00226 and ECE from 0.01587 ± 0.00346 to 0.02633 ± 0.00118.

Table 6. Out-of-fit calibration and fixed-threshold selection

Dataset	Raw Brier	Calibrated Brier	Raw ECE	Calibrated ECE	t*	Threshold range	Threshold-role F1
Farm-Flow	0.14307 ± 0.00844	0.15452 ± 0.00594	0.08739 ± 0.01576	0.09902 ± 0.01178	0.604 ± 0.026	0.583-0.633	0.88589 ± 0.00408
CICIoT2023	0.00587 ± 0.00012	0.00646 ± 0.00011	0.00181 ± 0.00024	0.00412 ± 0.00036	0.638 ± 0.100	0.534-0.734	0.99576 ± 0.00014
UNSW-NB15	0.06046 ± 0.00195	0.06155 ± 0.00226	0.01587 ± 0.00346	0.02633 ± 0.00118	0.336 ± 0.094	0.231-0.412	0.91736 ± 0.00288

The results therefore do not support presenting logistic calibration as a performance improvement. Its role is procedural: score mapping is fitted independently of threshold selection and final testing.

Selected-Latent versus Full-Latent Representation:-

The primary contextual comparison evaluates the selected-latent configuration against the recorded Random Forest reference operating on all 128 CNN-Lite dimensions.

On Farm-Flow, selected-latent F1 was **0.4724 ± 0.0131**, compared with **0.4550 ± 0.0058** for the full-latent reference. On CICIoT2023, the corresponding values were **0.9953 ± 0.0001** and **0.9949 ± 0.0001**. On UNSW-NB15, selected-latent F1 was **0.7480 ± 0.0183**, compared with **0.7578 ± 0.0045**.

Table 7. Contextual selected-latent versus full-latent F1

Dataset	Selected-latent RF	Full-latent RF
Farm-Flow	0.4724 ± 0.0131	0.4550 ± 0.0058
CICIoT2023	0.9953 ± 0.0001	0.9949 ± 0.0001
UNSW-NB15	0.7480 ± 0.0183	0.7578 ± 0.0045

The selected configuration is higher in the recorded Farm-Flow comparison, nearly identical on CICIoT2023, and lower on UNSW-NB15. These differences remain descriptive because the experimental budgets are not identical. The primary selected-latent RF uses 300 trees, whereas the recorded RF references use 150 trees; CICIoT2023 references additionally use capped 100,000-row cohorts.

The table therefore cannot establish that latent selection caused an improvement or preserved performance under an equal-budget ablation. It shows only that the sparse representations remained within the performance range of the available full-latent references under the recorded configurations.

DISCUSSION:-

Answer to the Research Question:-

The results provide a conditional answer to the research question. IWHO-Lite substantially reduced the 128-dimensional CNN-Lite representation on all three evaluated datasets. Mean retained dimensionality was 63.00 dimensions on Farm-Flow, 37.67 on CICIoT2023, and 41.67 on UNSW-NB15, corresponding to compression rates of 50.78%, 70.57%, and 67.45%.

The relatively narrow within-dataset ranges-59-65, 37-39, and 40-44-show that similarly sized subsets were repeatedly produced across the three model seeds. This observation does not imply that the same latent indices were recovered because independently trained CNN-Lite encoders can distribute information differently across their latent bases.

Whether useful clean detection remained after sparsification was more dependent on the dataset. Mean clean F1 was 0.4724 on Farm-Flow, 0.9953 on CICIoT2023, and 0.7480 on UNSW-NB15. The largest compression therefore coincided with the strongest F1 on CICIoT2023, whereas Farm-Flow retained substantially more coordinates yet remained the most difficult distribution.

IWHO-Lite should consequently be interpreted as a mechanism for explicitly searching the performance-parsimony space, not as a guarantee that reducing the latent representation will automatically preserve predictive quality.

Interpretation of Latent-Dimension Selection:-

The findings support the separation of representation learning from representation sparsification. CNN-Lite first constructs a nonlinear 128-dimensional latent space. IWHO-Lite then evaluates subsets of that space rather than altering the original input feature schema.

This distinction also limits how individual selected coordinates can be interpreted. Latent coordinate 17 in one independently trained encoder cannot be assumed to carry the same information as coordinate 17 in another. The more defensible cross-run quantities are therefore the number of retained dimensions, compression rate, objective value, clean detection metrics, and downstream inference characteristics.

The results likewise do not establish that IWHO-Lite is a globally optimal subset-search method or superior to alternative metaheuristics. Such a claim would require controlled equal-budget comparisons across competing optimizers. The evidence in this study establishes instead that the specified binary search procedure repeatedly identifies substantially reduced latent subsets under the recorded experimental design.

Farm-Flow as the Critical Agricultural Result:-

Farm-Flow is particularly important because it provides the closest match to the article's agricultural application domain. Its result should therefore not be obscured by the near-perfect positive-class performance observed on CICIoT2023.

IWHO-Lite reduced the Farm-Flow latent representation by approximately half, retaining 63.00 ± 3.46 dimensions. The final detector nevertheless achieved F1 0.4724 ± 0.0131. This was not caused by low attack sensitivity: recall remained 0.9523 ± 0.0189. Instead, precision was 0.3143 ± 0.0137 and specificity 0.3167 ± 0.0560, indicating a substantial false-positive burden.

This profile has direct operational significance. An agricultural IDS that detects most attacks but frequently classifies benign traffic as malicious may create unnecessary alerts, escalation, or intervention. High recall alone is therefore insufficient for judging practical utility.

The Farm-Flow result does not invalidate latent selection. It demonstrates an important boundary: **representation sparsification does not remove distribution-specific classification difficulty**. The result makes the article's performance-parsimony interpretation stronger because it prevents representation compactness from being conflated with operational effectiveness.

The current evidence does not identify one particular physical cause for the Farm-Flow false-positive profile. Explanations involving specific equipment, farms, seasons, synchronization mechanisms, or capture conditions would require additional metadata or experiments and are therefore not inferred here.

Dataset-Dependent Behavior on CICIoT2023 and UNSW-NB15:-

CICIoT2023 represents a very different operating regime. It retained the fewest dimensions and achieved the largest compression while maintaining F1 0.9953 ± 0.0001. Specificity, however, remained lower than its positive-class metrics, reinforcing the need to interpret the strongly attack-rich distribution beyond F1 alone.

UNSW-NB15 produced an intermediate profile. Its 41.67 ± 2.08 selected dimensions corresponded to 67.45% compression and F1 0.7480 ± 0.0183. Recall remained high, while precision and specificity were more moderate.

The three datasets therefore show that similar representation-reduction strategies can coexist with very different detection profiles. CICIoT2023 and UNSW-NB15 broaden the evaluation space, but they do not prove transfer to an agricultural environment. Farm-Flow remains the domain-specific anchor.

Interpreting the Selected-Latent versus Full-Latent Comparison:-

The selected-latent and full-latent references provide useful context for assessing whether the recorded sparse configurations operate in the same general performance range as the complete representation.

The selected configuration produced higher recorded F1 on Farm-Flow, nearly identical F1 on CICIoT2023, and lower F1 on UNSW-NB15. This pattern shows that substantial latent reduction was not accompanied by a uniformly adverse recorded F1 profile.

It does **not**, however, establish that IWHO-Lite improved or preserved performance causally. The primary selected-latent Random Forest uses 300 trees, whereas the reference Random Forest uses 150, and CICIoT2023 references additionally use capped cohorts. These budget differences prevent the comparison from functioning as a controlled ablation.

The defensible interpretation is therefore that the selected representations produced recorded F1 values of broadly similar order to the available full-latent references while retaining substantially fewer latent dimensions. A strict causal comparison requires equal-budget selected-versus-full experiments.

Latent Compression Is Not Resource Efficiency:-

The measured results provide a clear reason not to equate latent compression with complete resource efficiency.

CNN-Lite is compact, with 31,905 total parameters and an approximately 0.097 MiB serialized encoder. IWHO-Lite reduces the dimensionality presented to the final Random Forest by approximately 51-71%. These measurements support a claim of **representation parsimony**.

They do not determine the size of the final tree ensemble. Farm-Flow retained the largest latent subset but produced the smallest Random Forest at 5.59 ± 0.05 MiB. CICIoT2023 retained only 37.67 dimensions but produced a 39.57 ± 1.07 MiB forest, while UNSW-NB15 retained 41.67 dimensions and produced the largest forest at 57.07 ± 4.22 MiB.

The downstream ensemble structure consequently depends on more than the number of inputs. Training distribution, tree depth, split structure, and ensemble complexity also affect serialized size.

The same boundary applies to latency. Mean batch latency of 0.0973-0.1344 ms per sample characterizes the recorded software environment. It does not demonstrate electrical-energy consumption, peak RAM, battery autonomy, thermal suitability, or real-time behavior on a physical agricultural gateway.

The evidence therefore supports **latent-space compression with measured software-level inference characterization**, not hardware-validated resource efficiency.

Role of the Leakage-Aware Evaluation Design:-

The leakage-aware protocol supports the credibility of the performance-parsimony analysis by making the provenance of fitted decisions explicit. Fitted preprocessing remains within the training scope, model-development decisions precede calibration and threshold selection, and the final test population remains outside these choices. Exact-vector and contradictory-label audits further expose identifiable dataset artifacts rather than allowing them to remain hidden within aggregate metrics.

The protocol should nevertheless not be described as eliminating all leakage. Exact-vector auditing cannot detect every dependency arising from devices, farms, sessions, temporal capture structure, or future distribution drift. The released datasets also do not provide a harmonized device, farm, session, capture, or timestamp identifier suitable for applying the same grouped or temporal partition design across all three datasets.

Leakage-aware evaluation is therefore a **methodological support for the latent-selection analysis**, rather than a separate claim of absolute data independence.

Implications for AG-IoT Intrusion Detection:-

The results suggest that explicit latent-space sparsification can be incorporated into a modular AG-IoT intrusion-detection pipeline without assuming that every encoder output must be supplied to the terminal classifier. This provides a mechanism for making representation complexity itself part of model design.

The Farm-Flow findings simultaneously show why representation reduction must be evaluated alongside operational detection behavior. A compact representation with high attack recall may still be unsuitable if false alarms remain excessive. Future deployment-oriented work should therefore evaluate thresholds and classifier configurations against explicit costs of false positives and false negatives under the conditions of the target agricultural system.

The present fixed F1-oriented threshold provides a reproducible operating point but does not represent changing farm conditions, explicit asymmetric costs, or online distribution drift.

Limitations:-

The study has four main limitations.

First, experimental evidence is based on **three independent model seeds**. This is sufficient to expose descriptive inter-run variation but not to provide population-level uncertainty equivalent to a larger repeated-experiment design. Statements concerning repeated selection are therefore limited to the evaluated seeds.

Second, the available datasets do not provide a harmonized device, farm, session, capture, or timestamp identifier suitable for a common grouped or temporal split. Exact-vector overlap auditing removes direct duplicate vectors across roles but does not model distribution drift across farms, devices, seasons, capture periods, or future operating conditions. Individual latent coordinate identities are also seed-specific and are not aligned across independently trained encoders.

Third, the computational evaluation remains software-level. Selected dimensionality, serialized model size, batch latency, and throughput are measured, but electrical energy, peak RAM, thermal behavior, battery autonomy, accelerator utilization, and execution on an agricultural gateway are not.

Fourth, contextual reference systems use non-identical budgets. The primary Random Forest has 300 trees, whereas the recorded RF references have 150; CICIoT2023 references are additionally capped. The selected-versus-full comparison therefore cannot isolate the causal contribution of latent selection. Logistic mapping also did not improve aggregate Brier score or ECE, and the fixed threshold does not adapt to explicit deployment costs or distribution drift.

Future evaluation should prioritize equal-budget selected-versus-full ablations, larger repeated designs, grouped or temporal validation where metadata allow, direct agricultural-gateway memory and energy measurements, and operating-point selection tied to explicit false-positive and false-negative costs.

CONCLUSION:-

This study investigated whether a learned intrusion-detection representation can be substantially reduced before terminal classification while retaining a useful detection profile under leakage-aware evaluation. The proposed pipeline combines CNN-Lite representation learning, IWHO-Lite binary latent-dimension selection, and Random Forest classification, while logistic score mapping and fixed-threshold selection remain separate downstream decision stages.

Across three independent model seeds, IWHO-Lite retained**63.00 of 128 dimensions on Farm-Flow, 37.67 on CICIoT2023, and 41.67 on UNSW-NB15**, corresponding to compression rates of **50.78%, 70.57%, and 67.45%**, respectively. Mean clean-test F1-scores were **0.4724, 0.9953, and 0.7480**. These findings answer the research question conditionally: substantial latent reduction was repeatedly obtained, but the detection profile associated with the reduced representation remained strongly dataset-dependent.

Farm-Flow provides the most important domain-specific qualification. Its selected representation maintained high recall but produced a substantial false-positive burden, showing that latent compactness alone does not resolve agricultural traffic discrimination. Measured inference results reinforce the same conclusion from another perspective: fewer latent dimensions did not translate monotonically into smaller Random Forest models, and recorded batch latency is not equivalent to hardware-level efficiency.

The study therefore supports **latent-dimension selection as a performance-parsimony design strategy under leakage-aware evaluation**, rather than universal predictive superiority or hardware-validated resource efficiency. Future work should emphasize controlled equal-budget comparisons, broader repeated or grouped evaluation, direct agricultural-gateway measurements, and operating-point selection informed by explicit deployment costs.

REFERENCES:-

[1] M. Campoverde-Molina and S. Luján-Mora, "Cybersecurity in smart agriculture: A systematic literature review," *Computers & Security*, vol. 150, Art. no. 104284, 2025, doi: 10.1016/j.cose.2024.104284.

[2] D. Javeed, T. Gao, M. S. Saeed, and P. Kumar, "An intrusion detection system for edge-envisioned smart agriculture in extreme environment," *IEEE Internet of Things Journal*, vol. 11, no. 16, pp. 26866-26876, 2024, doi: 10.1109/JIOT.2023.3288544.

[3] A. Thakkar and R. Lohiya, "A survey on intrusion detection system: Feature selection, model, performance measures, application perspective, challenges, and future research directions," *Artificial Intelligence Review*, vol. 55, no. 1, pp. 453-563, 2022, doi: 10.1007/s10462-021-10037-9.

[4] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, and K. Rieck, "Dos and Don'ts of Machine Learning in Computer Security," in *31st USENIX Security Symposium (USENIX Security 22)*, Boston, MA, USA, 2022, pp. 3971-3988.

[5] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," *Patterns*, vol. 4, no. 9, Art. no. 100804, 2023, doi: 10.1016/j.patter.2023.100804.

[6] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman, "Leakage in data mining: Formulation, detection, and avoidance," *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 4, Art. no. 15, 2012, doi: 10.1145/2382577.2382579.

[7] A. K. Kanga, D. J. Diako, S. Oumtanaga, and Y. C. Brou, "Design of a CNN-DNN hybrid model optimized by IWHO for intrusion detection in smart agricultural networks: Evaluation on the Farm-Flow benchmark," *Journal of Information Systems Engineering and Management*, vol. 10, no. 56s, pp. 694-702, 2025.

[8] C.-W. Kuo and C.-X. Wu, "A lightweight two-stage intrusion detection framework optimized for edge-based IoT environments," *Computers, Materials & Continua*, vol. 87, no. 3, Art. no. 53, 2026, doi: 10.32604/cmc.2026.076767.

[9] A. Mazinani, D. Antonucci, L. Davoli, and G. Ferrari, "Performance assessment of deep learning for network intrusion detection on a constrained IoT device," *Future Internet*, vol. 18, no. 1, Art. no. 34, 2026, doi: 10.3390/fi18010034.

[10] R. Ferreira, I. Bispo, C. Rabadão, L. Santos, and R. L. de C. Costa, "Farm-flow dataset: Intrusion detection in smart agriculture based on network flows," *Computers & Electrical Engineering*, vol. 121, Art. no. 109892, 2025, doi: 10.1016/j.compeleceng.2024.109892.

[11] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," *Sensors*, vol. 23, no. 13, Art. no. 5941, 2023, doi: 10.3390/s23135941.

[12] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, ACT, Australia, 2015, pp. 1-6, doi: 10.1109/MilCIS.2015.7348942.

[13] G. T. Francis, A. Souri, and N. İnanç, "A systematic review of metaheuristic based feature selection strategies for cyber-attack detection in the IIoT," *Artificial Intelligence Review*, vol. 59, Art. no. 73, 2026, doi: 10.1007/s10462-025-11473-7.

[14] P. Vurubindi, J. Frnda, C. N. Sujatha, P. B. Divakarachari, G. S. Nijaguna, and A. Mahendar, "Optimizing feature selection with random reversal and adaptive Gaussian based Dung beetle optimizer for intrusion detection system in IoT," *Scientific Reports*, vol. 15, Art. no. 45314, 2025, doi: 10.1038/s41598-025-29278-7.

[15] S. Baek, J. Jeon, B. Jeong, and Y.-S. Jeong, "Hybrid meta-heuristic feature selection model for network traffic-based intrusion detection in AIoT," *Computer Modeling in Engineering & Sciences*, vol. 145, no. 1, pp. 1213-1236, 2025, doi: 10.32604/cmes.2025.070679.

[16] M. A. Abdel-Rahman, A. S. Alluhaidan, S. A. El-Rahman, A. E. Masnour, A. S. I. Amar, M. A. Sobh, A. M. Bahaa-Eldin, T. Shamseldin, and M. Shalaby, "Feature importance guided autoencoder for dimensionality reduction in intrusion detection systems," *Scientific Reports*, vol. 16, Art. no. 5013, 2026, doi: 10.1038/s41598-026-36695-9.

[17] M. Karthikeyan, R. Brindha, M. M. Vianny, V. Vaitheeshwaran, M. Bachute, S. Mishra, and B. B. Dash, "Integration of metaheuristic based feature selection with ensemble representation learning models for privacy aware cyberattack detection in IoT environments," *Scientific Reports*, vol. 15, Art. no. 22887, 2025, doi: 10.1038/s41598-025-05545-5.

[18] I. Naruei and F. Keynia, "Wild horse optimizer: A new meta-heuristic algorithm for solving engineering optimization problems," *Engineering with Computers*, vol. 38, no. S4, pp. 3025-3056, 2022, doi: 10.1007/s00366-021-01438-z.

[19] R. Zheng, A. G. Hussien, H.-M. Jia, L. Abualigah, S. Wang, and D. Wu, "An improved Wild Horse Optimizer for solving optimization problems," *Mathematics*, vol. 10, no. 8, Art. no. 1311, 2022, doi: 10.3390/math10081311.

[20] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large-Margin Classifiers*, A. J. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. Cambridge, MA, USA: MIT Press, 2000, pp. 61-74.
[21] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proceedings of the 22nd International Conference on Machine Learning (ICML '05)*, 2005, pp. 625-632, doi: 10.1145/1102351.1102430.
[22] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning*, PMLR, vol. 70, 2017, pp. 1321-1330.
[23] E. Effah, O. Thiare, and A. M. Wyglinski, "A tutorial on agricultural IoT: Fundamental concepts, architectures, routing, and optimization," *IoT*, vol. 4, no. 3, pp. 265-318, 2023, doi: 10.3390/iot4030014.
[24] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5-32, 2001, doi: 10.1023/A:1010933404324.