Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 09 , September - 2026

Quantum Network Intrusion Detection System: A Controlled Evaluation with Ablated Controls

K K Vishvaa, Abhay Sharma, Davana P Lad, Dr B P Pradeep Kumar*,
Department of Computer Science and Engineering, Atria Institute of Technology, Bengaluru, Karnataka,

***Abstract*—**Variational quantum circuits are said to buy richer decision boundaries for fewer parameters than a classical network needs, and intrusion detection on network flows is among the tasks where that claim is made most often. What backs it up is thinner than the publication count suggests: hybrid quantum IDS papers typically show one training run against loosely specified baselines, leaving no way to separate an architectural effect from run-to-run variance. We report a controlled evaluation of a hybrid quantum-classical detector, Q-NIDS, across three configurations sharing one preprocessing pipeline. Two use CICIDS2017 with a six-qubit circuit combining data re-uploading, a learnable entanglement topology and quantum recurrent memory gates; the third scales to eight qubits on BigFlow-NIDS, a 66.9-million-flow NetFlow corpus, using a batched state-vector simulator matching PennyLane's `lightning.qubit` to 9.7×10^{-8}. On CICIDS2017 the calibrated model reaches 87.97% accuracy and 49.09% macro F1, exceeding a 1D-CNN ($40.52 \pm 0.83\%$), an LSTM ($39.06 \pm 2.04\%$) and an MLP ($37.98 \pm 3.97\%$) trained on identical data, but falling 12.47 points short of a Random Forest ($61.56 \pm 0.20\%$). Three findings cut against the usual story. Most of the margin over the neural baselines is bought by post-hoc prior adjustment, a classical correction worth 8.11 macro-F1 points. The learnable entanglement topology never trains: a non-differentiable branch in gate selection cuts its gradient path, after which an unopposed entropy regulariser drives the adjacency matrix to a value where every conditional gate switches off. On BigFlow-NIDS an ablated control with the quantum layer deleted reaches 10.79% macro F1 against the full model's 12.53%, a margin too narrow to credit to quantum processing. We report all of it, failed configurations included, and argue that controls of this kind belong in any hybrid quantum machine learning claim.

***Index Terms*—**quantum machine learning, variational quantum circuit, network intrusion detection, class imbalance, logit adjustment, ablation study.

I. Introduction

Two hard problems meet in network intrusion detection. One is volume: a single day of capture on even a modest testbed yields hundreds of thousands of flow records [1],[3]. The other is rarity — web attacks make up 102 of the 424,190 test flows in the CICIDS2017 corpus used throughout, or 0.024% of the data [4],[5]. Label everything benign and you clear 80% accuracy while detecting nothing. Macro-averaged F1 weights every attack class equally however few examples it has, so that is the number we treat as meaningful, and headline accuracies here should be read with suspicion [7],[8].

Quantum machine learning is one proposed route out of this. A variational quantum circuit maps classical inputs into a Hilbert space whose dimension grows exponentially with qubit count, so a few dozen parameters can in principle express decision boundaries a classical network would need thousands of weights to reproduce [2], [6]; data re-uploading strengthens the claim, since injecting the input repeatedly between variational blocks makes even a single-qubit circuit a universal approximator [9],[10]. Should those arguments hold in practice, a compact quantum classifier trained on a small balanced sample ought to generalise to rare attack classes better than a classical model of similar size [11],[2].

Whether they hold is a different question, and the published record leaves it open [13],[14]. One pattern recurs: the quantum model is shown once, with no seed variance, next to classical baselines that either carry variance or have barely been tuned[15],[16]. Most parameters usually live in the classical read-out stacked on the circuit, and its size often goes unreported [17],[18]. No control is run with the quantum layer removed [19],[20]. A positive result obtained under those conditions cannot be interpreted — the circuit may be doing the work, or the read-out may be, or the run may have been a favourable draw [21],[22].

We took the opposite approach. Q-NIDS was built with the features the literature suggests should help: data re-uploading across a temporal window, a learnable rather than fixed entanglement topology, and quantum recurrent gates between time steps [23],[24]. It was evaluated three times under deliberately different conditions, preprocessing held fixed, classical baselines run over three seeds on identical inputs, and two ablated controls added in the third configuration [25],[26]. What follows is what we found, including the parts that did not work [27],[28].

Our contributions are a controlled three-configuration evaluation sharing one circuit family and read-out design; an accounting of where performance originates, separating circuit from read-out and from post-hoc calibration, which alone supplies 8.11 macro-F1 points; a batched state-vector simulator agreeing with PennyLane's C++ backend to 9.7×10^{-8} while cutting per-sample cost from 20.9 ms to 0.467 ms; a negative result on learnable entanglement topology traced to a reproducible cause; and a BigFlow-NIDS audit finding 45.59% duplicate feature vectors and destination-port purity high enough to constitute testbed leakage.

II. Related Work

A. Classical machine learning for intrusion detection

Flow-based intrusion detection has long been dominated by tree ensembles and, more recently, deep sequence models. Random forests [29] remain a strong default: flow features are largely axis-aligned in their discriminative structure, since thresholds on packet counts, durations and inter-arrival times separate attack families cleanly, and bagged trees degrade gracefully under imbalance [30],[31]. CICIDS2017 [1] is the reference benchmark, with labelled captures across five weekdays. Reported accuracies routinely exceed 99%, which invites caution: most come from random shuffled splits allowing flows from one attack burst into both training and test sets, and chronological splitting reduces them substantially. Independent analyses also report labelling and feature-generation defects in the released CSV files [32].

B. Variational quantum circuits

The variational quantum circuit is the workhorse of near-term quantum machine learning [33], [34], [35]: data is encoded into qubit rotations, a parameterised sequence of rotations and entangling gates is applied, and expectation values of chosen observables become a differentiable function of the parameters. Gradients follow analytically on hardware through the parameter-shift rule ; in simulation, adjoint differentiation gives the exact gradient at a cost independent of parameter count. Two design choices dominate. Schuld et al. [3] showed the representable functions are fixed by the frequency spectrum the encoding gates induce, so repeating the encoding as in data re-uploading [4] widens the accessible function class, while

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 09 , September - 2026

McClean et al. [5] showed gradient variance in random circuits decays exponentially with qubit count, producing barren plateaus.

C. Hybrid quantum models and the research gap

Applying variational circuits to intrusion detection raises an obstacle image classification avoids: flow records carry dozens of features while simulable circuits have a handful of qubits. Nearly all hybrid designs therefore insert a classical compression stage, commonly an autoencoder, ahead of the quantum layer — with a consequence rarely discussed. Compression network, circuit and read-out are all trainable, so unless the read-out is deliberately constrained it can absorb the entire classification task, leaving the circuit an elaborate but inert intermediate representation. Quantum recurrent designs insert learnable gates between time steps so one flow influences the next [23]; quantum convolutional [24] and quantum kernel [7] methods offer alternative routes.

Table I summarises how representative work handles the methodological questions that decide whether a positive result can be interpreted. Architectural novelty is usually reported carefully; seed variance, read-out parameter accounting and quantum-removed controls are usually absent. Our aim is to supply exactly those missing elements for one concrete architecture, and to report the outcome whether or not it flatters the method.

TABLE I
Methodological Practices in Representative Hybrid Quantum Classification Studies, and the Position Taken in This Work

Aspect	Common practice in the literature	Why it limits interpretation	Approach taken here
Reporting of the quantum model	Single training run, no variance	A favourable seed cannot be distinguished from a real effect	Reported as a single run and labelled as such; baselines run over three seeds
Classical baselines	Often untuned, sometimes quoted from other papers	Comparison is against a straw model, not the best classical option	Four baselines trained on identical inputs with the same loss and early stopping
Read-out capacity	Rarely reported separately from circuit parameters	The classical head may be performing the classification	Parameter split reported; two read-out widths compared directly
Quantum-removed control	Usually absent	No evidence that the circuit contributes anything	Two controls in Configuration C: circuit deleted, and circuit frozen at random initialisation
Train/test partitioning	Random shuffled split	Flows from one attack burst leak across the split	Chronological split within contiguous per-file, per-class blocks
Evaluation distribution	Balanced or resampled test set	Overstates rare-class performance relative to deployment	Test set left at its natural distribution (80.3% benign)

III. METHODOLOGY

A. System architecture

Fig. 1 shows the pipeline, in four stages. Stage A cleans and partitions raw flow records and hands on scaled feature vectors. Stage B brings those down to a dimensionality the circuit will take, then rescales them into a legal rotation-angle range. Stage C is the quantum circuit; it reads a short temporal window and emits one expectation value per class. Stage D turns those into class scores, calibrates them against the deployment class prior, and hosts the post-hoc analyses. Keeping B and C apart was deliberate: with the autoencoder trained first and frozen, the circuit works against a fixed input representation throughout its own training, removing one source of instability and making its contribution easier to isolate.

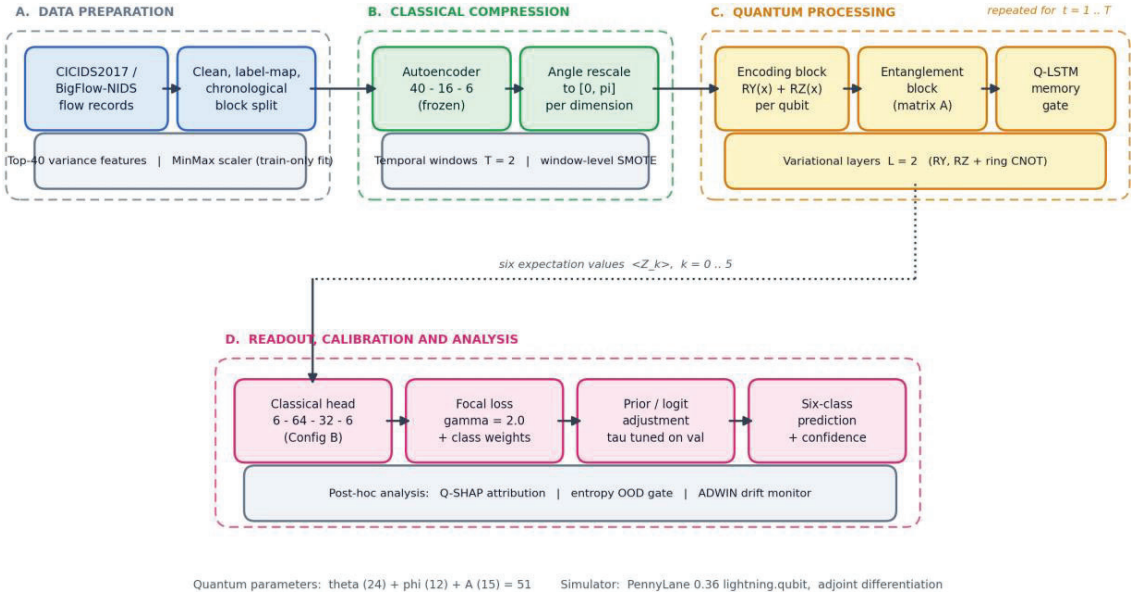

Fig. 1. End-to-end Q-NIDS architecture. Stage A performs cleaning, chronological block partitioning, variance-based feature selection and train-only scaling. Stage B compresses 40 features to 6 through a frozen autoencoder and rescales them to $[0, \pi]$. Stage C applies the quantum circuit over a temporal window of $T = 2$ flows. Stage D contains the classical read-out, the imbalance-aware loss, prior adjustment, and the explainability and drift-monitoring components. Dotted lines carry the six measured expectation values from the circuit to the read-out.

B. Datasets

CICIDS2017. The eight daily capture files were read individually rather than concatenated, preserving within-file row order corresponding to capture order. Removing rows with non-finite values left 2,827,829 flows across 78 numeric features. Fine-grained labels were folded into six operational categories: Normal, DoS, PortScan, BruteForce, Botnet and WebAttack. Table II gives the resulting distribution, which spans nearly four orders of magnitude between the most and least frequent class.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 09 , September - 2026

TABLE II
CICIDS2017 Class Distribution After Label Mapping and Chronological Partitioning

Class	Train	Validation	Test	Test share
Normal	1,589,921	340,693	340,706	80.32%
DoS	265,815	56,959	56,963	13.43%
PortScan	111,162	23,820	23,822	5.62%
BruteForce	10,736	2,300	2,303	0.54%
Botnet	1,369	293	294	0.07%
WebAttack	471	100	102	0.02%
Total	**1,979,474**	**424,165**	**424,190**	**100%**

BigFlow-NIDS. Configuration C uses BigFlow-NIDS [25], [26], assembled from four NetFlow benchmark collections (NF-UNSW-NB15-v3 [20], NF-ToN-IoT-v3, NF-BoT-IoT-v3, NF-CSE-CIC-IDS2018-v3) under the harmonised feature standard of Sarhan et al. [27]. It holds 66,935,021 flows over 55 attributes with 32 attack categories. Our profiling reproduced the flow count exactly and identified three capture eras: 2015 (2,365,424 flows, 3.53%), 2018 (37,049,337, 55.35%) and 2019 (27,520,260, 41.11%). Several labels occur in exactly one era — DDOS_attack-HOIC and FTP-BruteForce only in 2018, xss and password only in 2019 — confirming eras correspond to distinct source collections, which matters for split design.

C. Preprocessing and leakage control

Four decisions were made specifically to avoid optimistic bias. Rather than shuffling, each (file, class) pair was treated as one contiguous chronological run and split 70/15/15 in time order, with a block identifier carried to the windowing stage so a window can never span two runs; test flows are therefore later in capture order than training flows of the same class and file. Feature variance for selection, the MinMax scaler, the autoencoder weights and the encoder output range were computed on the training partition alone. SMOTE [9] was applied only to training windows and only after the split, so no synthetic sample can derive from a validation or test flow. For Configuration C, both IP address and both port fields were dropped, since destination port alone carries a majority-class purity of approximately 0.83 on the profiled subset — the fixed port assignment of a staged testbed, not a generalisable property of the attack.

D. Dimensionality reduction and angle encoding

Forty selected features exceed what a six-qubit circuit accepts, so a shallow undercomplete autoencoder [22] compresses them, with a 40 → 16 → 6 ReLU encoder and mirrored decoder. Training minimises reconstruction error over the full real training partition,

$$L_{\mathrm{AE}} = (1/N)\,\Sigma_i \,\| x_i - g(f(x_i)) \|^2_2 \tag{1}$$

where f and g denote encoder and decoder. After 50 epochs of Adam [21] at learning rate 10^{-3}, reconstruction MSE reached 8.4×10^{-5}, at which point the autoencoder was frozen. Encoder outputs are then rescaled into $[0, \pi]$, the range over which one rotation traverses the Bloch sphere without wrapping. Writing z for the encoder output and $z_{\min}$, $z_{\max}$ for per-dimension extrema on the training partition,

$$\tilde{x}_j = \pi \cdot (z_j - z_{\min,j}) / (z_{\max,j} - z_{\min,j} + \varepsilon) \tag{2}$$

E. Quantum circuit design

Fig. 2 shows the circuit, which processes a window of T consecutive flow vectors through four gate groups. In the *encoding block*, each qubit receives a Y- and a Z-rotation parameterised by the corresponding encoded feature at time step t:

$$U_{\mathrm{enc}}(\tilde{x}(t)) = \otimes_{q=0}^{n-1} R_Z(\tilde{x}_q(t))\, R_Y(\tilde{x}_q(t)) \tag{3}$$

Repeating this at every time step realises data re-uploading [4], so the input enters the circuit T times rather than once. The *entanglement block* uses a learnable symmetric matrix A to decide which qubit pairs couple: the continuous matrix is sigmoid-squashed, symmetrised, zeroed on the diagonal and binarised at 0.5, with a CNOT applied wherever the binarised entry is one,

$$A_{\mathrm{cont}} = \sigma(A_{\mathrm{logit}}),\quad A_{\mathrm{sym}} = \tfrac{1}{2}(A_{\mathrm{cont}} + A_{\mathrm{cont}}^{\mathrm{T}}) - \mathrm{diag}(\cdot),\quad A_{\mathrm{bin}} = [A_{\mathrm{sym}} > 0.5] \tag{4}$$

Since the indicator has zero derivative almost everywhere, a straight-through estimator [16] passes the incoming gradient unchanged through binarisation in the backward pass; Section V-G shows this was not sufficient and explains why. Between consecutive time steps a *quantum memory gate* — a learnable rotation pair per qubit, parameterised by φ — carries information forward:

$$U_{\mathrm{mem}}(\varphi(t)) = \otimes_q R_Z(\varphi_{q,1}(t))\, R_Y(\varphi_{q,0}(t)) \tag{5}$$

Once all time steps are absorbed, L *variational layers* apply a parameterised Y- and Z-rotation on every qubit followed by an unconditional ring of CNOTs connecting qubit q to $(q+1)$ mod n. This ring matters for interpreting our results: being fixed rather than learned, it leaves the circuit entangling capacity regardless of what the learned topology does. Finally, the Pauli-Z expectation value of each of the first C qubits is measured, giving one real value per class:

$$h_k = \langle \psi_{\mathrm{out}} | Z_k | \psi_{\mathrm{out}} \rangle \in [-1, 1],\quad k = 0, \ldots, C-1 \tag{6}$$

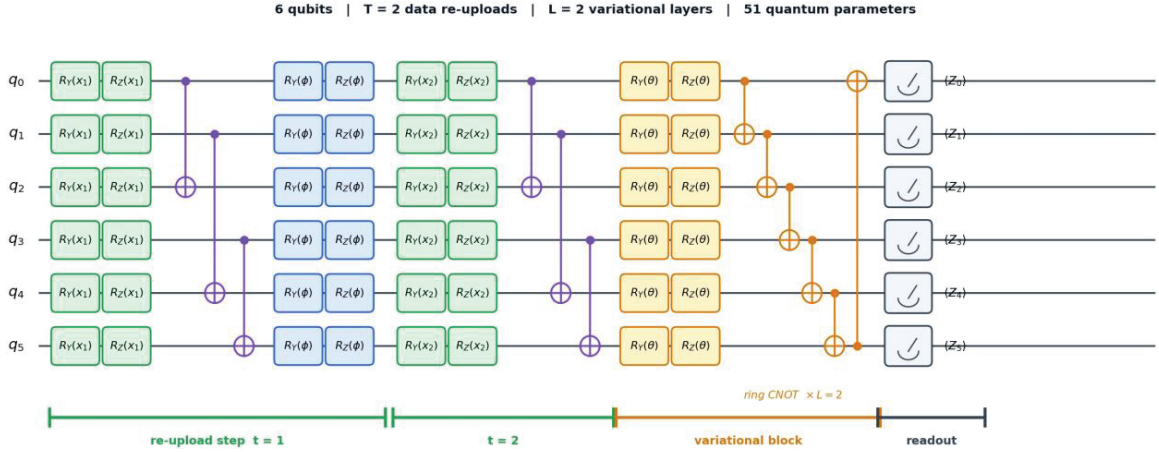

Fig. 2. Q-NIDS circuit for n = 6 qubits, T = 2 re-uploads and L = 2 variational layers. Green gates encode the flow features at each time step; purple gates are the conditional CNOTs selected by the learned adjacency matrix, of which an illustrative subset is drawn; blue gates are the quantum memory rotations between time steps;

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 09 , September - 2026

amber gates form the variational block with its fixed CNOT ring. Measurement returns six Pauli-Z expectation values. The rotation parameters number 24 (θ) plus 12 (φ), with a further 15 unique entries in the topology matrix.

F. Read-out, loss and prior adjustment

The expectation values pass to a small classical head. Two widths were compared: a narrow 6 → 16 → 6 variant, and the wider 6 → 64 → 32 → 6 head used in Configurations A and B, with ReLU activations and dropout 0.2, contributing 2,726 parameters against the circuit's 36 rotation parameters and 36 topology entries. We report this split explicitly because it bounds how much of the model's behaviour can be attributed to the circuit. Training minimises a class-weighted focal loss [10], which down-weights well-classified examples and so concentrates gradient on the rare attack classes:

$$L_{\text{focal}} = -\Sigma_c \, w_c \, (1 - p_c)^{\gamma} \, y_c \log p_c, \quad \gamma = 2.0, \quad w_c = N / (C \cdot N_c) \tag{7}$$

The model trains on a class-balanced window set but is evaluated on the natural distribution, in which 80.3% of flows are benign, and that mismatch systematically inflates rare-class scores at test time. We correct it post hoc using logit adjustment [17], subtracting a scaled log-prior from each class score,

$$\tilde{h}_c = h_c - \tau \cdot \log \pi_c \tag{8}$$

where π_c is the class prior from the training partition and $\tau \in [0, 1]$ controls correction strength, selected by grid search on a natural-distribution validation subset of 50,000 windows and never on test.

G. Experimental configurations

Table III defines the three configurations. A and B differ only in class-balancing regime and training budget, isolating the effect of oversampling policy. C changes dataset, qubit count, simulator and read-out simultaneously, so it is a scale-up study rather than a controlled variation, and we treat it as such throughout.

TABLE III
The Three Experimental Configurations

Property	Config A	Config B	Config C
Dataset	CICIDS2017	CICIDS2017	BigFlow-NIDS
Classes	6	6	26 (grouped from 32)
Qubits	6	6	8
Re-uploads / blocks	T = 2	T = 2	4 blocks
Observables	6 (Z only)	6 (Z only)	44 (Z, ZZ, X)
SMOTE policy	capped at 3× real count	full, to 1000 per class	none (per-class sampling cap)
Training windows / flows	5,705 windows	6,000 windows	442,506 flows
Simulator	PennyLane lightning.qubit	PennyLane lightning.qubit	custom batched state vector
Read-out	6–64–32–6	6–64–32–6	linear, no hidden layer
Ablated controls	none	none	no-quantum, frozen-circuit
Training time	22.3 min	80.6 min	133.7 min

IV. IMPLEMENTATION

A. Environment and training protocol

Configurations A and B ran on PyTorch 2.6.0 (CPU) with PennyLane 0.36.0 [8] and its `lightning.qubit` C++ backend under adjoint differentiation, chosen over parameter-shift because its cost does not scale with parameter count: parameter-shift would need roughly 612,000 circuit evaluations per epoch against about 12,000 for adjoint. JAX was pinned at 0.4.25 to avoid an incompatibility between newer releases and PennyLane 0.36; it is not used at runtime. Supporting libraries were scikit-learn 1.3.2, imbalanced-learn 0.11.0 and river 0.21.0. Configuration C ran on the same PyTorch version with the simulator of Section IV-B. Q-NIDS was optimised with Adam at learning rate 0.01, batch size 32, gradient-norm clipping at 5.0 and at most 50 epochs, with early stopping on validation macro-F1 at patience 10. Parameters were initialised at 0.1 × N(0, 1), since smaller initialisation placed the model near a flat region of the loss surface.

Model selection deserves comment, since optimistic bias often enters there. We capped the validation set per class, giving a roughly balanced set of 1,396 windows on which macro-F1 is a meaningful selection signal; an earlier randomly sampled subset held no WebAttack windows at all, leaving two of six classes invisible to the criterion. Validation macro-F1 is therefore computed on a balanced set and is not comparable to test macro-F1 on the natural distribution: it is a selection signal, not a performance estimate.

B. Batched state-vector simulation

The per-sample QNode call dominates cost in Configurations A and B: we clocked it at 20.9 ms per window, consistent with the 94 s per epoch observed. Across 442,506 training flows that becomes prohibitive, so for Configuration C we wrote a state-vector simulator holding an entire mini-batch as one tensor and applying gates as batched tensor contractions, with gradients from PyTorch autograd — the same exact gradients adjoint differentiation gives. Correctness was checked against PennyLane 0.36 `lightning.qubit` on the A/B circuit: maximum absolute deviation was 9.7×10^{-8}, consistent with float32 rounding, and state norms stayed within [0.999997, 1.000002]. CPU throughput came to 1,481.4 µs per flow at batch size 256 and 467.0 µs at batch size 1,024, roughly a 45× reduction against the per-sample loop with the numerical result unchanged.

C. Ablated controls

Configuration C adds two controls sharing encoder, loss, optimiser and linear read-out with the full model, differing only in what sits between encoder and read-out. In *Control 1 (no quantum)* the circuit is deleted and the encoder output feeds the read-out directly; should this match the full model, the circuit contributes nothing. In *Control 2 (frozen circuit)* the circuit is retained with parameters left at random initialisation, a fixed random feature map; should this match the full model, training the circuit contributes nothing whatever the circuit itself does. The Configuration C read-out is one linear layer with no hidden units, so any nonlinear decision boundary must come from the encoder or the circuit, not the head — the constraint that makes the comparison informative.

V. RESULTS

A. Training dynamics

Fig. 3 shows Configuration B's training curve. The loss falls smoothly from 1.43 to roughly 0.34 over 48 epochs, while validation macro-F1 climbs steeply through the first ten epochs then creeps upward with substantial epoch-to-epoch movement,

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 09 , September - 2026

peaking at 0.8929 on epoch 38 before early stopping ended training at epoch 48 (80.6 minutes total). Configuration A looked nothing like this: its validation macro-F1 swung wildly under the capped-SMOTE regime (0.096, 0.396, 0.351, 0.754, 0.612, 0.194), and the run stopped at epoch 14 with epoch 4 supplying the best checkpoint. Picking a checkpoint off a curve that noisy is close to drawing at random among recent epochs. Configuration B's fully balanced training set gave a much steadier trajectory, and that stability — not any change to the circuit — is the clearest single difference between the runs.

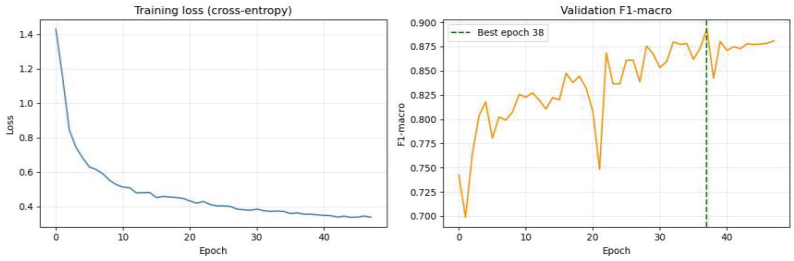

Fig. 3. Configuration B training dynamics. Left: focal training loss over 48 epochs. Right: validation macro-F1 measured on the class-balanced validation subset, with the selected checkpoint at epoch 38 marked. Validation here is balanced by construction and is used for model selection only; it is not comparable to the natural-distribution test scores in Tables IV to VI.

B. Test performance and the effect of calibration

Table IV reports Q-NIDS on the CICIDS2017 test set of 212,091 windows at its natural class distribution, under both configurations and both before and after prior adjustment. Three things stand out. The balancing regime matters more than anything else we varied: moving from capped to full SMOTE lifted macro-F1 from 30.31% to 40.98% with an identical circuit. Prior adjustment behaves very differently across configurations — in A it raised accuracy by 27 points while reducing macro-F1, simply relabelling uncertain predictions as benign, whereas in B it improved both, adding 11.00 accuracy points and 8.11 macro-F1 points, indicating the underlying probability ranking was sound and only miscalibrated. Per-class thresholding tuned on the balanced validation set was actively harmful, costing 4.04 macro-F1 points, because thresholds fitted on a balanced distribution do not transfer to an imbalanced one.

How large that calibration effect is deserves a plain statement. Of the 8.57-point margin by which Configuration B exceeds the strongest neural baseline, 8.11 points arrive from a post-hoc classical correction involving no quantum computation at all.

TABLE IV
Q-NIDS Test Performance on CICIDS2017 (212,091 Windows, Natural Distribution). All Values Are Single Runs.

Configuration and decision rule	Accuracy	Macro F1	Macro prec.	Macro rec.
Config A — argmax	53.63%	30.31%	34.22%	62.83%
Config A — per-class thresholds	56.87%	26.27%	31.47%	54.82%
Config A — prior adjusted ($\tau = 0.15$)	80.86%	29.52%	33.45%	38.09%
Config B — argmax	76.97%	40.98%	38.81%	82.14%
Config B — prior adjusted ($\tau = 0.25$)	**87.97%**	**49.09%**	**47.27%**	**59.63%**

C. Comparison with classical baselines

Four classical baselines were trained on identical windowed inputs with the same loss, class weights and early-stopping criterion, each over three seeds. Table V and Fig. 4 give the comparison.

TABLE V
Q-NIDS (Config B) Against Classical Baselines on the CICIDS2017 Test Set. Baselines Are Mean ± SD Over Three Seeds; Q-NIDS Is a Single Run.

Model	Accuracy	Macro F1	Macro prec.	Macro rec.
MLP (no temporal context)	71.21 ± 9.06%	37.98 ± 3.97%	38.90 ± 1.73%	74.16 ± 7.53%
LSTM	75.38 ± 5.03%	39.06 ± 2.04%	38.87 ± 2.13%	76.23 ± 0.80%
CNN-1D	77.89 ± 1.04%	40.52 ± 0.83%	39.85 ± 0.92%	80.77 ± 1.13%
Q-NIDS (ours, adjusted)	**87.97%**	**49.09%**	**47.27%**	**59.63%**
Random Forest	94.42 ± 0.09%	61.56 ± 0.20%	56.55 ± 0.28%	87.35 ± 0.08%

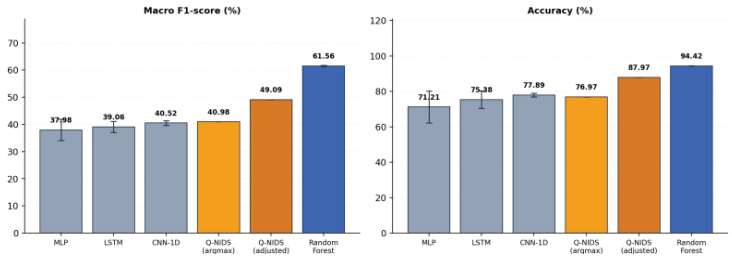

Fig. 4. Macro F1 (left) and accuracy (right) on the CICIDS2017 test set. Grey bars are neural baselines with three-seed error bars; amber bars are Q-NIDS before and after prior adjustment; the blue bar is the Random Forest. Q-NIDS bars carry no error bar because they represent single runs.

Q-NIDS exceeds all three gradient-trained neural baselines on both metrics, and its 8.57-point macro-F1 margin over the CNN-1D sits well outside that baseline's seed spread of ±0.83. Against the Random Forest it falls 12.47 macro-F1 points and 6.45 accuracy points short; the forest trains in about two seconds on the same 6,000 windows, against 80.6 minutes.

More informative is that every gradient-trained model here loses heavily to the forest: the MLP, LSTM and CNN cluster between 37.98% and 40.52% macro-F1 while the forest reaches 61.56% on identical inputs. Whatever disadvantages this pipeline imposes are therefore not specific to the quantum

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 09 , September - 2026

model; they affect the whole gradient-trained family, and Q-NIDS sits at its favourable end rather than outside it.

D. Per-class behaviour

Table VI decomposes performance by class, and Fig. 5 gives the corresponding confusion matrix for the uncalibrated Configuration B model. Q-NIDS leads on Normal, PortScan, BruteForce and WebAttack, and the BruteForce margin is large: 0.44 against 0.104 for the CNN. It loses on DoS. Botnet is a complete failure — prior adjustment drove the class to zero recall, and every one of the 147 Botnet windows went to Normal.

We would rather state that failure than explain it away. Before adjustment the model did recall all 147 Botnet windows, at precision 0.01, which meant flagging 11,229 benign windows as Botnet. Neither operating point is usable. The difficulty underneath is arithmetic: where prevalence is 0.07%, even 99.9% specificity on the benign class yields 212 false positives against 147 true positives, capping precision at 0.41. Every model in Table VI struggles here, and the Random Forest's 0.172 is the best any of them achieve.

TABLE VI
Per-Class F1 on the CICIDS2017 Test Set. Q-NIDS Is Configuration B After Prior Adjustment; Baselines Are Seed 42.

Class	Support	Q-NIDS	CNN-1D	LSTM	MLP	Random Forest
Normal	170,350	0.93	0.872	0.891	0.858	0.967
DoS	28,481	0.67	0.709	0.716	0.722	0.872
PortScan	11,911	0.80	0.617	0.585	0.661	0.916
BruteForce	1,151	0.44	0.104	0.237	0.175	0.629
Botnet	147	0.00	0.032	0.042	0.019	0.172
WebAttack	51	0.11	0.041	0.045	0.027	0.149

Fig. 5. Configuration B confusion matrix on the full CICIDS2017 test set before prior adjustment (argmax decision rule). Rows are true classes. The dominant error mode is benign traffic distributed across the attack classes, with 11,229 benign windows predicted as Botnet and 8,757 as BruteForce. This is the behaviour prior adjustment subsequently corrects.

E. Effect of the read-out width

A diagnostic run compared read-out widths on an otherwise identical circuit. The narrow head (roughly 214 parameters) performed markedly worse than the wider head (2,726 parameters) used in Configurations A and B. The finding cuts both ways: the circuit's six expectation values carry more class information than a small head can extract, but 2,726 of the model's 2,798 parameters, or 97.4%, are classical. Any claim that the circuit drives performance must contend with that ratio, which is why Configuration C constrains the read-out to a single linear layer.

F. Configuration C: scale-up and controls

Configuration C applied the design to BigFlow-NIDS at eight qubits with a 44-observable measurement set (Table VII). The absolute figures are low, and the cause lies in the split rather than the model. Because attack campaigns in this corpus are themselves time-localised, splitting chronologically inside each capture era left 16 of the 26 retained classes missing from either the training or the test partition: *password* drew 26,383 training flows and none in validation or test, while DDoS drew 9,626 training flows against 41,524 test flows. A class absent from training has structurally zero F1, pulling the macro average down regardless of model quality. Configuration C therefore yields no usable estimate of achievable performance on BigFlow-NIDS, and we do not offer it as one.

What it does yield is the control comparison, which survives the split defect because both models face exactly the same partition. Macro F1 is 12.53% for the full quantum model against 10.79% for the no-quantum control — 1.74 points, from single runs, with no variance estimate. We do not take that as evidence the quantum layer contributes. The accuracy gap is wider, 30.31% against 17.26%, but accuracy on a 26-class problem dominated by a handful of frequent classes is a weak signal. The frozen-circuit control, which would have separated the contribution of having a circuit from that of training one, did not complete; we report it as missing rather than inferring its outcome.

Profiling BigFlow-NIDS produced two findings relevant to anyone else using it. Among 1,161,924 sampled flows, 529,770 were exact duplicates at the feature-vector level, a rate of 45.59%. The documentation describes a Spark deduplication step, but it evidently operated on full rows including addresses and timestamps, which does not remove records identical in every field a model actually sees; distributed across a split, such duplicates are a direct leakage channel, so we removed them before training. Separately, destination port alone predicts the majority class with purity approximately 0.83, an artefact of staged capture where each attack tool runs against a fixed service port. Even after dropping port and address fields, a hash-based audit reported roughly 97.8% overlap between train and test feature hashes. Hash collisions inflate that figure and it is not a leakage rate, but it is high enough that we flag it as unresolved and treat Configuration C's absolute numbers with corresponding caution.

TABLE VII
Configuration C on BigFlow-NIDS (94,826 Test Flows, 26 Classes). Single Runs.

Model	Trainable params	Accuracy	Macro F1	Macro precision	Macro recall
Q-NIDS v3 (full)	5,066	30.31%	12.53%	14.98%	16.50%
Control 1 — no quantum layer	3,826	17.26%	10.79%	15.17%	16.67%
Control 2 — frozen random circuit	run incomplete; not reported				

G. The learned entanglement topology does not train

We intended the learnable adjacency matrix as one of the design's contributions. It does not work, and its failure mechanism is specific enough to be worth documenting. Fig. 6 shows the topology matrix after Configuration B training: the continuous matrix A_{sym} is uniform across all off-diagonal entries, sitting essentially at its initial value, and the binarised matrix is entirely zero. In the selected checkpoint the entanglement block contributed no CNOT gates whatsoever. Configuration A, trained under a different regime, kept 11 of the 15 possible connections. When a parameter settles on "all connections" in one run and "no connections" in another, and task performance moves opposite to the connection count while it does so, that parameter is not responding to the task.

Inspecting the gradient confirms as much. CNOTs are chosen by a Python conditional evaluated on the binarised matrix, and that branch never enters the autograd graph. Bridging exactly this discontinuity was the point of the straight-through estimator, but it only substitutes a surrogate gradient

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 09 , September - 2026

for the binarisation function; it cannot restore a gradient path control flow already severed before the circuit was built. No signal from the classification loss reaches A_{logit}. That leaves the topology entropy regulariser, which rewards high binary entropy in A_{cont}, maximised at exactly 0.5. As the sole gradient source acting on the parameter it pushes every entry toward 0.5, and since binarisation applies a strict inequality, an entry sitting exactly at 0.5 produces no gate. The all-zero topology in Fig. 6 is thus fully explained: an unopposed regulariser drove the matrix to the precise value at which the thresholding rule switches everything off.

This has two consequences. Our claim that Q-NIDS learns a task-adapted entanglement topology is unsupported, and we withdraw it. The model is not thereby classically trivial: the variational block applies an unconditional CNOT ring at every layer, so the selected checkpoint still contains twelve two-qubit gates and generates entanglement. What Configuration B demonstrates is a data re-uploading circuit with fixed ring entanglement, and Tables IV to VI should be read as belonging to that architecture.

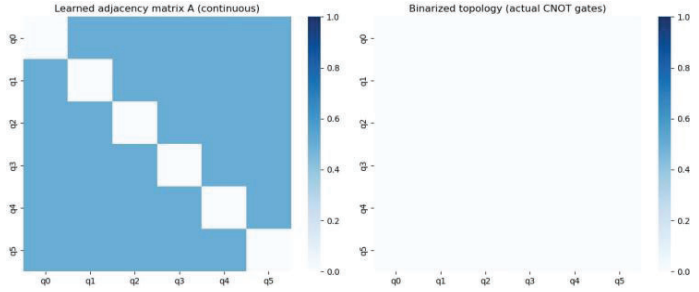

Fig. 6. Topology matrix after Configuration B training. Left: the continuous matrix A_{sym}, uniform across all off-diagonal entries at approximately its initialisation value. Right: the binarised matrix, entirely zero, meaning no conditional CNOT gates were applied. The uniformity is the signature of a parameter driven only by its entropy regulariser, with no gradient reaching it from the classification loss.

H. Auxiliary components

Three supporting components were evaluated on Configuration A. We report them as functional demonstrations rather than validated capabilities, since none was benchmarked against ground truth.

Feature attribution. A Shapley-value procedure [12] adapted to operate over circuit inputs was applied to individual flagged samples. Fig. 7 shows attributions for three DoS windows, all classified correctly. The pattern repeats across the three: encoded dimensions 2 and 4 push toward the DoS decision while dimension 1 pushes against it. Consistency across samples of one class is encouraging, but three samples is an anecdote, not evidence.

Out-of-distribution gating. Shannon entropy of the predicted distribution served as a novelty signal, with the threshold set at the 95th percentile of validation entropy, giving 2.3777. On the test stream, 13,106 of 212,091 windows (6.2%) exceeded it. Without labelled novel attacks we cannot say whether those flagged windows are genuinely anomalous.

Drift detection. ADWIN [11] was run over the prediction stream and reported 48 change points. Their distribution is uneven, with several clustering tightly beyond sample 193,000, which is consistent with the block structure of the test partition rather than with genuine concept drift. Establishing whether the detector tracks drift or partition boundaries requires drift-labelled data we do not have.

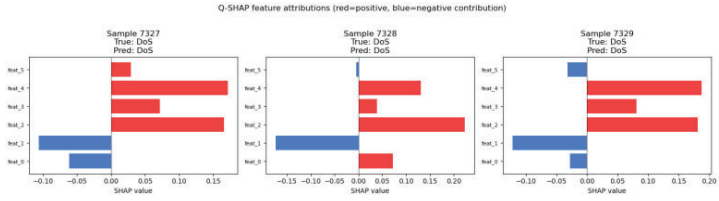

Fig. 7. Shapley attributions over the six encoded input dimensions for three DoS windows from Configuration A, all correctly classified. Red bars indicate contributions toward the predicted class, blue against. Dimensions 2 and 4 contribute positively in all three cases and dimension 1 negatively, suggesting a stable attribution pattern for this class, on a sample of three.

VI. Discussion

A. What the quantum circuit contributes

The defensible summary is this: on this task Q-NIDS is a competitive member of the gradient-trained family and an uncompetitive one against tree ensembles, and where it sits inside that family cannot be cleanly attributed to quantum processing.

The evidence pulls both ways. In favour, on identical inputs Configuration B beats three classical neural architectures by margins outside their seed spread, using 36 rotation parameters against the CNN's orders of magnitude more, with a wide advantage on BruteForce, a genuinely hard minority class. Against, 97.4% of its parameters are classical; the largest identified contributor to its score is a post-hoc calibration step with no quantum content; one of its three architectural novelties demonstrably never trained; and in the one configuration with an ablated control, removing the quantum layer cost just 1.74 macro-F1 points in a comparison never replicated. Cleaner attribution would take the frozen-circuit control Configuration C failed to finish, run over several seeds — in our view the single most important experiment still outstanding.

B. Why the tree ensemble wins

The Random Forest's 12.47-point lead is not a matter of tuning effort — it was trained on default settings in roughly two seconds. Something representational is the likelier explanation. Attack families are separated in flow features by threshold conditions on packet counts, durations and flag ratios, exactly the structure axis-aligned recursive partitioning captures outright while smooth parameterised functions can only approximate. Supporting that reading, the MLP, LSTM, CNN and quantum model all sit 20 or more points below the forest

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 09 , September - 2026

despite quite different inductive biases: the gap tracks the model family, not the individual model. The practical implication is more useful than a quantum-advantage claim would be. For flow-based intrusion detection over engineered features the default should be a tree ensemble, and a hybrid quantum architecture proposing to replace one takes on the burden of beating it, not of beating a neural baseline that already loses to it.

C. Threats to validity

Every Q-NIDS figure here comes from one training run. Three-seed variance accompanies the baselines and not the quantum model, an asymmetry weakening the comparison in our own favour; given the epoch-to-epoch variation visible in Fig. 3, the reported figures should be read as one draw from a distribution whose width we have not measured. $T = 2$ provides minimal temporal context, and because windows form within per-class blocks no window ever spans an attack boundary, so the memory gates have little to carry and the temporal claim is correspondingly weak. Configurations A and B train on about 6,000 windows out of 989,734 available, a restriction simulation cost forced on us; whether the Table V ranking survives at full scale is untested, though the Random Forest reaches 61.56% on those same 6,000 windows, suggesting the restriction is not obviously what binds. The chronological split assumes CSV row order reflects capture order, never independently checked because the numeric-dtype filter excluded the timestamp column. Configuration C changed dataset, qubit count, observables, simulator and read-out at once, so no part of the B-to-C difference can be pinned on a single factor. All results come from noiseless simulation; finite sampling, gate infidelity and decoherence would each degrade these figures by an unmeasured amount. Finally, CICIDS2017 has documented labelling and feature-generation issues [15], so results on it should not be extrapolated to production traffic without independent validation.

VII. Conclusion and Future Work

Across three configurations sharing one preprocessing pipeline we evaluated a hybrid quantum-classical intrusion detection architecture, reporting failures alongside successes. The calibrated model reaches 87.97% accuracy and 49.09% macro F1 on CICIDS2017, exceeding an LSTM, a 1D-CNN and an MLP trained on identical data while staying 12.47 macro-F1 points behind a Random Forest that trains in two seconds. One engineering contribution supported the rest: a batched state-vector simulator agreeing with PennyLane to 9.7×10^{-8} and cutting per-sample cost roughly 45-fold, without which the large-scale configuration would not have been feasible.

The negative findings are the useful ones. The learnable entanglement topology never trained, for a traceable reason: a hard branch in gate selection severed its gradient path, leaving an entropy regulariser free to drive it to precisely the value where the thresholding rule disables every gate. Post-hoc prior adjustment, a step with no quantum computation in it, accounted for 8.11 of the model's macro-F1 points, more than its entire margin over the strongest neural baseline. And in the single configuration carrying an ablated control, removing the quantum layer altogether cost 1.74 macro-F1 points in one unreplicated comparison. Taken together these do not support a claim of quantum advantage on this task, and we make none.

Four directions follow: replicate every configuration over at least five seeds and finish the frozen-circuit control, the experiment that would settle whether training the circuit matters; rebuild the topology mechanism as something differentiable end to end, swapping conditional CNOT selection for continuously parameterised two-qubit rotations; repair the Configuration C split so every retained class appears in training and test alike, and run the classical baselines the current run lacks; and validate a subset of the circuit on real hardware to quantify what finite sampling and gate noise cost in practice.

More broadly, control experiments of the kind reported here ought to be routine rather than remarkable. A hybrid quantum model keeping 97% of its parameters in a classical read-out, evaluated with no ablated control, cannot support conclusions about quantum contribution however good its headline number looks. The controls are inexpensive; running them changes what the result means.

References

[1] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy (ICISSP)*, 2018, pp. 108–116.

[2] Shrinivasa, P. K., Swamy, M., Kalegowda, B., & Puttappa, P. K. B. (2026). A bibliometric analysis of explainable artificial intelligence (XAI): Trends, themes, and global research dynamics. Journal of Scientometric Research, 15(2), 331–346. https://doi.org/10.5530/jscires.20260179

[3] Pramod K.B. Rangaiah, Robin Augustine,Leveraging machine learning for personalized type 2 diabetes prediction: A comparative analysis,Array,Volume 31,2026,101031,ISSN 2590-0056,https://doi.org/10.1016/j.array.2026.101031.

[4] G., P., R., S., & H. P., S. (2026). Quantitative Learning Impact Modelling of AI-Integrated Module-Level Project-Based Learning in a Multimodal Data Omics Course: A Case Study. Journal of Engineering Education Transformations, 39(Special Issue 2), 21–29. https://doi.org/10.16920/jeet/2026/v39is2/26003

[5] Mahadevaswamy, N., S., B., J., & Ninawe, S. S. (2026). Real-Time AI-Assisted Error Feedback System and Copilot for MATLAB Programming in Core Engineering Courses: A Comparative Analysis. Journal of Engineering Education Transformations, 39(Special Issue 2), 1–11. https://doi.org/10.16920/jeet/2026/v39is2/26001

[6] S., M., M. V., S.& G., P. (2026). Uncovering the Hidden Layers of Thinking: Extended Computational Thinking (CT) Strategies in Problem-Based Classrooms Learning. Journal of Engineering Education Transformations, 39(Special Issue 2), 12–20. https://doi.org/10.16920/jeet/2026/v39is2/26002

[7] M. V., S., N., S., P. Y., N., (2026). AI-Augmented Complexity Learning: Design, Automation, and Learning Impact for Conceptual Mastery in Derandomization Through Intelligent Tutoring and Real-Time Feedback. Journal of Engineering Education Transformations, 39(Special Issue 2), 30–39. https://doi.org/10.16920/jeet/2026/v39is2/26004

[8] Shanthamallappa, M., Basavaiah, J. et al. Deep neural network and hidden markov model hybrid automatic speech recognition system for recognizing spontaneous Kannada proverb's with perspectives on the future of ASR research. Multimed Tools Appl 85, 220 (2026). https://doi.org/10.1007/s11042-026-21167-z

[9] Naresh, E., Raghavendra, C.K. et al. Comprehensive brain tumour concealment utilizing peak valley filtering and deeplab segmentation. Sci Rep 15, 34780 (2025). https://doi.org/10.1038/s41598-025-18574-x

[10] Abdullah, R.Y., Venkatesan, C., Naresh, E. et al. AI driven hybrid convolutional and transformer based deep learning architecture for precise lung nodule classification. Sci Rep (2026). https://doi.org/10.1038/s41598-025-34569-0

[11] Sheela, S., Jyothi, S. & kumar, A Hybrid Approach for Vehicular Spectrum Allocation Using Artificial Neural Networks with Autoencoders and CSI Feedback. SN COMPUT. SCI. 6, 877 (2025). https://doi.org/10.1007/s42979-025-04413-3

[12] Pradeep Kumar, B.P. A Comparison of CNN, RNN, and FNN Algorithms to Investigate Effective Diabetes Prediction. Arch Computat Methods Eng (2025). https://doi.org/10.1007/s11831-025-10467-6

[13] Ravikumar J, Shankar B B, Manjunath Kamath K,Optimized anti-interference dynamic integral neural network approach for dementia prediction in health care,Knowledge-Based Systems, Volume 321, 2025, 113723, ISSN 0950-7051, https://doi.org/10.1016/j.knosys.2025.113723

[14] P. K. B. Rangaiah, Augustine, "From Microwave Measurement to Application: Enhancement of Fat-Intrabody Communication by Advanced Computational Techniques," 2025 16th German Microwave Conference (GeMiC), Dresden, Germany, 2025, pp. 490-493, doi: 10.23919/GeMiC64734.2025.10979071

[15] Kumar, B. P. (2018). analysis of ASL recognition system for aligned RGB-D image. journal of advanced research in dynamical and control systems, (1).

[16] Maaz Ahmed, Chapter 21 - IoT-based battery management system in the electric vehicle charging station, Editor(s): Tuan Anh Nguyen, Digital Twin, Blockchain, and Sensor Networks in the Healthy and Mobile City, Elsevier, 2025, Pages 431-451, ISBN 9780443341748, https://doi.org/10.1016/B978-0-443-34174-8.00022-3

[17] B. D J, G. G and M. N, "Fake Job Post Detection Using Machine Learning," 2025 International Conference on Computing for Sustainability and Intelligent Future (COMP-SIF), Bangalore, India, 2025, pp. 1-6, doi: 10.1109/COMP-SIF65618.2025.10969867.

[18] S. R. Chand, U. Ahmed, T. M. Ismail, S. B. Chand and C. P, "Email Validator Dashboard: A Comprehensive Tool for Accurate Email Validation," 2025 International Conference on Computing for Sustainability and Intelligent Future (COMP-SIF), Bangalore, India, 2025, pp. 1-8, doi: 10.1109/COMP-SIF65618.2025.10969888

[19] R. Kumar J, A. Biswas, B. B. Hafeeza and D. Dhaanya, "Emotion Classification in

Text Using Machine Learning and LSTM," 2025 International Conference on Computing for Sustainability and Intelligent Future (COMP-SIF), Bangalore, India, 2025, pp. 1-7, doi: 10.1109/COMP-SIF65618.2025.10969881

[20] R. M. Khan, S. Kabir and Y. Raj, "Driver Drowsiness Detection System," 2025 3rd International Conference on Smart Systems for applications in Electrical Sciences (ICSSES), Tumakuru, India, 2025, pp. 1-6, doi: 10.1109/ICSSES64899.2025.11010064

[21] S. S. Hari Gokul, A. Gow and L. Adarsh, "Remote Monitoring for Water Level of Bridges and Flood Zones," 2025 International Conference on Knowledge Engineering and Communication Systems (ICKECS), Chickballapur, India, 2025, pp. 1-6, doi: 10.1109/ICKECS65700.2025.11035864.

[22] S. Monishaa, S. Menon and S. A. Aiman H, "Real-Time Speech-to-Speech Translator: Analysis and Implementation," 2025 4th International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballari, India, 2025, pp. 1-6, doi: 10.1109/ICDCECE65353.2025.11035154.

[23] A. John, k. Harshitha and D. Pooja, "Leveraging Histopathological images for detection and diagnosis of Colonic Carcinoma," 2025 International Conference on Knowledge Engineering and Communication Systems (ICKECS), Chickballapur, India, 2025, pp. 1-6, doi: 10.1109/ICKECS65700.2025.11035505

[24] M. Tabassum, R. Neha and A. Khanum, "ChargeSmart: Effortless EV Charging with Real- Time Slot Booking," 2025 International Conference on Knowledge Engineering and Communication Systems (ICKECS), Chickballapur, India, 2025, pp. 1-6, doi: 10.1109/ICKECS65700.2025.11035724.

[25] S. R, R. R and S. Shetty, "Universal Readability: Simplification of Complex Text Using Deep Learning," 2025 3rd International Conference on Smart Systems for applications in Electrical Sciences (ICSSES), Tumakuru, India, 2025, pp. 1-7, doi: 10.1109/ICSSES64899.2025.11009498.

[26] R. Purushotham, R. Mahato and S. M. V. Kumar, "AI-Powered Plagiarism Detection: A Web-Based System for Text Integrity," 2025 International Conference on Knowledge Engineering and Communication Systems (ICKECS), Chickballapur, India, 2025, pp. 1-6, doi: 10.1109/ICKECS65700.2025.11036044.

[27] R. Afnar, R. Raheem, N. Ranjanyou, and S. Veena, "Hand Motions Based Virtual AI Mouse," 2025 International Conference on Knowledge Engineering and Communication Systems (ICKECS), Chickballapur, India, 2025, pp. 1-6, doi: 10.1109/ICKECS65700.2025.11035082

[28] M. Harshvardhan, M. Devineni and P. A. Raju, "Framework for Tuberculosis Detection," 2025 International Conference on Knowledge Engineering and Communication Systems (ICKECS), Chickballapur, India, 2025, pp. 1-7, doi: 10.1109/ICKECS65700.2025.11035324

[29] L. G. Reddy, M. Chethana and D. Likhita, "Improving DDos Attack Identification and Mitigation In SDN Through The Use of Ensemble Online Machine Learning Framework," 2025 International Conference on Knowledge Engineering and Communication Systems (ICKECS), Chickballapur, India, 2025, pp. 1-6, doi: 10.1109/ICKECS65700.2025.11035808

[30] M. S, J. S. Jakati, C. Umarani, "Deep Learning Methods for Liver Tumor Segmentation: An Extensive Investigation and Analysis," 2025 1st International Conference on Advancement in Futuristic Technologies (ICAFT), Belagavi, India, 2025, pp. 1-8, doi: 10.1109/ICAFT66710.2025.11452940

[31] M. S, C. Umarani, S. S. Ittannavar, B. Alias Pratima Khot, "Automated Deep Fake Video Detection using Modified Semi Supervised Learning," 2025 1st International Conference on Advancement in Futuristic Technologies (ICAFT), Belagavi, India, 2025, pp. 1-7, doi: 10.1109/ICAFT66710.2025.11452690

[32] S. S. Ittannavar, R. Hebbale, M. S and B. Alias Pratima Khot, "Multi-Band Noise-Adaptive DEMON with SNR-Weighted Fusion for Bearing Fault Diagnosis," *2026 4th International Conference on Knowledge Engineering and Communication Systems (ICKECS)*, CHICKABALLAPURA, India, 2026, pp. 1-6, doi: 10.1109/ICKECS70176.2026.11528006.

[33] S. R, J. S. Jakati, B. Alias Pratima Khot, M. S , "Degradation-Aware Gearbox Fault Diagnosis Using LOFARgram-Based Band Descriptors and Run-Level Fusion," *2026 4th International Conference on Knowledge Engineering and Communication Systems (ICKECS)*, CHICKABALLAPURA, India, 2026, pp. 1-6, doi: 10.1109/ICKECS70176.2026.11528162

[34] J. S. Jakati, B. A. P. Khot, M. S , "Robustness-Oriented Explainability Analysis of the PHQ-9 Depression Scale Using Ensemble Learning Under Response Perturbations," *2026 4th International Conference on Knowledge Engineering and Communication Systems (ICKECS)*, CHICKABALLAPURA, India, 2026, pp. 1-6, doi: 10.1109/ICKECS70176.2026.11527572

[35] M. S, R. H. Havaldar, S. S. Ittannavar, V. R. S , "A Hybrid Classical–Quantum Variational Learning Framework for Breast Mass Classification," *2026 4th International Conference on Knowledge Engineering and Communication Systems (ICKECS)*, CHICKABALLAPURA, India, 2026, pp. 1-6, doi: 10.1109/ICKECS70176.2026.11527499.