Journal of Theoretical and Applied Information Technology
15st September 2026. Vol.104. No.17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

EDGECROP: A LIGHTWEIGHT SQUEEZE-AND-EXCITATION-GUIDED ATTENTION CASE-NET FOR CROP WATER STRESS CLASSIFICATION ON EDGE MICROCONTROLLERS

DEEPIKA KATARIA[1], RAMESH KUMAR C[2]

[1]Research Scholar, School of Computer Science and Engineering, Galgotias University, Greater Nodia, Uttar Pradesh, India.

[2]Professor, School of Artificial intelligence (SoAI), Galgotias University, Greater Nodia, Uttar Pradesh, India.

E-mail: [1]todeepika1991@gmail.com, [2]toramesh83@gmail.com

ABSTRACT

Autonomous irrigation decision-making at the field edge remains constrained by a persistent trade-off between classification accuracy and deployability on low-cost microcontrollers, particularly across the diverse agro-climatic conditions of Indian smallholder agriculture. This paper introduces EdgeCrop, a benchmark for crop water stress inference constructed from FAO-56 soil water balance simulation across multiple crops and agro-climatic zones, and evaluates sixteen candidate architectures spanning classical machine learning, convolutional, recurrent, and hybrid attention-based models. Motivated by the observation that no existing architecture jointly performs channel-level feature selection and temporal attention within a footprint suitable for microcontroller deployment, we propose CASE-Net, a hybrid architecture comprising a convolutional encoder, a squeeze-and-excitation recalibration block, and a self-attention encoder arranged sequentially so that channel recalibration conditions the features subsequently attended to by the attention mechanism. A full-capacity variant and a compressed variant, related through a capacity-transfer training procedure, are evaluated alongside the fifteen baseline models using classification accuracy, post-training quantization behaviour, simulated microcontroller latency and energy, and downstream irrigation-scheduling performance relative to a calendar-based baseline. Results show that the proposed architecture achieves leading classification accuracy at multiple compression tiers, with the compressed variant occupying a favourable accuracy-versus-footprint position relative to competing lightweight architectures and remaining within the energy and memory budgets of representative solar-powered microcontroller targets. A quantization anomaly specific to attention-based models is identified and explained. The scheduling simulation further reveals that classification accuracy alone is insufficient to predict irrigation-scheduling performance, underscoring the need to evaluate proposed architectures using both classification and downstream operational metrics rather than classification accuracy in isolation.

Keywords: *TinyML; Edge Computing; Crop Water Stress Inference; Irrigation Scheduling; Squeeze-And-Excitation Networks; Self-Attention; Knowledge Distillation; Capacity Transfer; FAO-56 Water Balance Model; Microcontroller Deployment; Precision Agriculture; Model Compression; Quantization; Smart Irrigation; Indian Agro-Climatic Zones*

1. INTRODUCTION

India's agriculture is the largest freshwater consumer in the country, with three major crops rice, wheat and sugarcane occupying only 40% of Indian gross cropped area yet consuming more than 80% of the irrigation water available nationally [1].

On most smallholder farms, irrigation scheduling continues to follow rigid calendar-based intervals of 5–10 days without reference to in-field soil-moisture status, monsoon progression or FAO-56 water-stress coefficients. Two simultaneous inefficiencies result: (a) declining cereal yield stability driven by heat-wave and drought conditions, compounded with groundwater-table declines that directly constrain future crop productivity in Punjab and Haryana [2]; and (b) over-application during monsoon months, which drives nitrogen leaching, waterlogging and persistent basin-level aquifer depletion [2]. Figure 1. Limitation of calendar-based irrigation scheduling and the proposed EdgeCrop three-class decision framework. (Left) The fixed seven-day irrigation interval practised by Indian smallholder farmers produces crop water stress and yield loss when root zone depletion exceeds the readily available water threshold during dry spells (red zone), and wasteful over-irrigation and waterlogging when precipitation has already recharged the soil profile (blue zone). (Right), The proposed EdgeCrop framework replaces the fixed calendar with a real-time three-class inference decision derived from the FAO-56 water

Journal of Theoretical and Applied Information Technology
15st September 2026. Vol.104. No.17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

stress coefficients executed continuously on a low-cost field microcontroller at 15-minute duty cycle intervals.

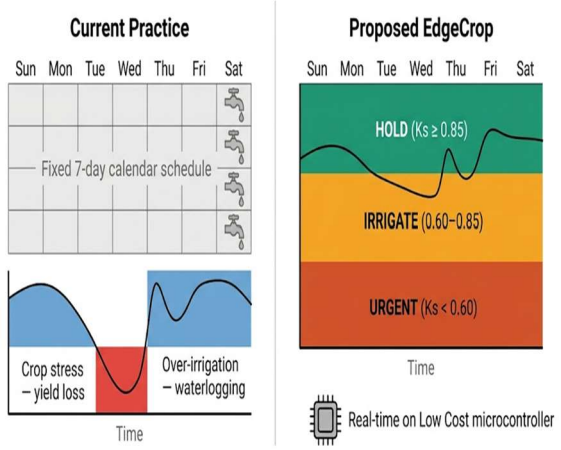

Figure 1: These are the two failure modes of calendar-driven scheduling against the three-class real-time decision framework proposed in this work

Recent reviews confirm that machine learning can deliver competitive accuracy for irrigation scheduling decisions, with deep learning increasingly outperforming process-based FAO-56 simulations on daily Ks tasks [3]. Deployment at the farm edge, however, imposes a different design envelope: typical agricultural MCUs offer 256 KB–8 MB flash, 20–512 KB SRAM, and solar-harvested power budgets supporting only ~15-minute inference duty cycles per day. A 2026 deployment-oriented review of low-cost TinyML in precision agriculture confirms that ESP32, STM32 and ATMega platforms dominate the on-device inference ecosystem, with approximately half of surveyed agricultural TinyML studies employing INT8 post-training quantization, and structured pruning, multi-objective compression and hardware-aware neural architecture search remaining comparatively under-researched [4]. Despite this hardware maturity, no published benchmark yet guides practitioners on the accuracy-deployment trade-off for multi-crop, multi-zone Indian irrigation conditions [4].

The TinyML community has assembled a mature compression toolkit INT8 post-training quantization, structured pruning, low-rank decomposition, lightweight network design and knowledge distillation that has emerged as a key enabler of on-device inference by aligning naturally with the integer arithmetic units available on microcontrollers [5]. Application of these techniques to agronomic time-series irrigation, however, remains limited: existing ML deployments target a single crop in a single agro-climatic zone, are constrained by limited data availability and reproducibility, and rarely couple compression-toolkit evaluation with multi-crop field data. A 2024 framing of the same gap identifies hardware cost, sensor uncertainty, data ownership and end-user trust as unresolved barriers between AI's promises and its real-world deployment at the farm edge [6]. This paper addresses these gaps through three primary contributions. First, we present the EdgeCrop benchmark a reproducible TinyML irrigation dataset covering five Indian crops (rice, wheat, maize, cotton, sugarcane), five agro-climatic zones and four growing seasons from 2018 through 2021 yielding 14,610 labelled daily records stratified into three operational classes: hold ($Ks \geq 0.85$), irrigate ($0.60 \leq Ks < 0.85$) and urgent ($Ks < 0.60$). Second, we propose CASE-Net, a hybrid CNN-Transformer architecture that places squeeze-and-excitation channel recalibration between the convolutional encoder and the multi-head temporal attention encoder, focusing attention exclusively on physically meaningful soil-water channels before temporal dependencies are computed. CASE-Net applies a reduction ratio of 4, suppressing 10 categorical feature channels and amplifying four critical soil-water channels (swc, Ks, ETo, ETc). CASE-Net Prime achieves macro-F1 = 1.0000 at 48,083 parameters; CASE-Net Lite achieves macro-F1 = 0.9288 at 5,963 parameters and 34.2 KB INT8. The highest F1 of any sub-40 KB model in the 16-architecture benchmark. Third, we specify a two-tier deployment strategy. CASE-Net Prime targets server-class and WiFi-connected nodes (ESP32-S3, STM32L4). CASE-Net Lite targets direct field deployment on the ₹150 RP2040 MCU, sustaining 3.555 ms inference latency and 262.89 mJ/day on STM32L4 well within the harvest envelope of a 1 W solar panel under Indian insolation across all five zones.

The remainder of this paper is organised as follows. Section 2 reviews related work and identifies eight research gaps. Section 3 describes the EdgeCrop dataset and preprocessing pipeline. Section 4 details the sixteen model architectures including CASE-Net. Section 5 presents the TFLite compression pipeline. Section 6 reports the MCU deployment benchmark. Section 7 reports classification results. Section 8 evaluates irrigation planning impact. Section 9 discusses deployment recommendations. Section 10 addresses limitations and future work, and Section 11 concludes.

2. RECENT WORK

This paper synthesises evidence from five intersecting research streams: machine learning for irrigation scheduling, TinyML and edge artificial intelligence for precision agriculture, model compression for microcontroller deployment, hybrid attention architectures for agronomic time-series, and irrigation planning evaluation. Across these streams, eight specific research gaps motivate the present work.

2.1 Machine Learning for Irrigation Scheduling

A 2024 systematic review of sixteen ML irrigation studies identified three persistent field-scale constraints limited data availability combined with data sharing restrictions, the absence of uncertainty quantification,

and the need for physics-informed ML that prevent reproducibility and cross-study comparison [3]. An extensive bibliometric analysis confirms that India ranks highest globally with 33 documents and 1,497 citations on ML for irrigation management [7], yet the bulk of these works remain single-crop, single-zone deployments calibrated to a private dataset.

Process-based FAO-56 simulations remain the de facto benchmark against which ML is evaluated. In the US Midwest, an XGBoost irrigation framework validated against the FAO-56 Spreadsheet and SETMI in 2024 field trials predicted soil-water depletion with RMSE 22–24 mm ($R^2 \geq 0.72$), with ML-generated depths (62–70 mm for maize; 73–83 mm for soybean) aligning with FAO-56 within ±4 mm [8]. Comparable recurrent baselines have emerged for long-horizon forecasting: an LSTM trained on a three-year vineyard record for southwest Idaho achieved a cross-validation mean MSE of 0.18, significantly outperforming a linear-regression baseline at MSE 1.29 [9]. A unified predictive scheduler combining k-means zoning, LSTM soil-moisture prediction and a reinforcement-learning agent trained via proximal policy optimisation recorded 7–23% water savings and 10–35% irrigation-water-use-efficiency gains on a 26.4-hectare Lethbridge field across the 2015 and 2022 growing seasons [10].

A 2025 deep-learning irrigation decision-making review identifies a persistent forecast-to-schedule gap: high ETo, root-zone soil moisture or ETc prediction accuracy does not translate end-to-end into executable multi-day schedules [11]. Indian ML work has progressed in parallel but along narrower trajectories. XGBoost ETo modelling at ICAR-IARI New Delhi using a 31-year (1990–2020) dataset confirmed reproducibility for semi-arid India [12]. A 2026 IoT-enabled precision-irrigation management system integrating Random Forest cloud ETo prediction with a fuzzy-logic decision engine demonstrated a 31.4% water-use reduction in Telangana rice plus a water-use-efficiency gain from 0.41 to 0.67 kg grain/m³ (+63.4%) [13]. A regional AIoT deployment combining soil-moisture sensing with edge-cloud scheduling reported 30–50% water savings across Punjab, Tamil Nadu and Maharashtra field scenarios with explicit forecast-to-schedule coupling [14]. Hybrid CNN-LSTM with multi-head attention and skip connections trained on Indian wheat and rice achieved RMSE 0.017, MAE 0.09 and R^2 0.967 [15].

2.2 TinyML and Edge AI for Precision Agriculture

A 2025 survey traces the trajectory from Tiny Machine Learning to Tiny Deep Learning, noting that TinyML devices typically operate below 1 mW with 32–512 kB SRAM, often without hardware floating-point accelerators or operating systems [16]. A 2026 in-depth review confirms that the field centres on three optimisation levers pruning, post-training quantization and low-rank factorisation alongside hardware-aware co-design, with deployment on ultra-low-power MCUs remaining constrained by joint computer-memory budgets [17]. The 2026 deployment-oriented precision-agriculture review reports that ESP32, STM32 and ATMega dominate the inference ecosystem; approximately half of surveyed agricultural TinyML works employ INT8 PTQ, while structured pruning, multi-objective compression and hardware-aware neural architecture search remain under-researched. The same review notes that explicit flash, RAM, MAC, latency and millijoule-level energy metrics are inconsistently reported [4].

Earlier comprehensive surveys corroborate this hardware ecosystem: 2023 TinyML taxonomy documents applications across smart farming, healthcare, environment and anomaly detection [18], and a 2023 reformable-TinyML survey explicitly identifies scarce benchmarking tools as a key obstacle to deployment readiness [19]. Empirical deployment studies confirm that quantised CNN models on ESP32 can exceed 87% accuracy for crop-field object classification at low energy cost [20]. FLOPs-Aware Knowledge Distillation deployed on the ESP32's 1 MB PSRAM lifted PlantVillage accuracy from 92.77% to 96.55% with INT4 quantization [21]. A 2026 lightweight-attention framework for multimodal sensor fusion on ESP32-S3, STM32H7 and nRF52840 demonstrates 12.3% higher accuracy than MobileNetV3 with 3.2× lower memory and 2.8× lower latency, enabling real-time fusion below 2 mW [22]

2.3 Model Compression for TinyML

A 2025 hardware-centric MCU quantization survey synthesises the optimisation toolkit pruning, low-rank decomposition, lightweight network design, neural architecture search and knowledge distillation and concludes that quantization is a key enabler of TinyML by aligning naturally with integer arithmetic units on MCUs [5]. A 2025 adaptive PTQ framework demonstrates up to 65% size reduction and 45% inference-latency improvement with under 10% accuracy drop, via fine-grained layer-wise sensitivity profiling and mixed INT8/INT4 representations on ARM Cortex-M4 and Cortex-M7 [23]. Teaching only the integer pathway that INT4 devices can reproduce, FLOPs-Aware Knowledge Distillation combined with INT4 yielded a deployable MobileNetV2 student on ESP32 with preserved PlantVillage accuracy [21].

Adaptive-temperature mixed-sample knowledge distillation, transferring from a MobileNetV3-Large teacher to compact residual CNN students, exceeded 96.7% accuracy across three student configurations on a Rosa damascena maturity dataset; the compact (0.75× width, 1.3 M parameters) student reached 97.11% accuracy at 72.19 ms inference latency [24]. A 2026 PRISMA-guided review of 92 KD studies in smart agriculture introduces the Performance Retention Ratio

as a standardised cross-domain metric, identifying adaptive, hardware-aware and explainable distillation as emerging directions [25]. For cotton-disease recognition, spot-adaptive distillation from DenseNet40 to ShuffleNetV2 yielded 90.59% accuracy with FLOPs reduced from 0.29 G to 0.045 G, a 6.4× compression [26].

2.4 Hybrid and Attention Architectures for Agronomic Time Series

A BiLSTM-CNN-Attention model for spring-corn irrigation volume in Jilin combined convolutional spatial feature extraction with bidirectional recurrence and an attention layer, achieving R^2 0.9749 on the test set [27]. A CNN + LSTM + multi-head self-attention system has been validated for rainfall prediction and fruit-health monitoring in dense-population, agriculture-dependent regions [28]. A 2024 transformer-versus-LSTM comparison for soil-moisture prediction in shallow-groundwater-level areas highlights attention interpretability as an open question, noting that recurrent architectures' vanishing gradients and serial computation make them unsuited to long-horizon forecasting [29].

Outside irrigation, attention-hybrid methodologies have matured. A CWT + Squeeze-and-Excitation + Transformer architecture achieved 97.4% accuracy and 97.1% macro-F1 on the KU-HAR benchmark [30]. A TSEBG architecture placing SE before BiGRU combining temporal convolutional encoding with bidirectional-gated-recurrence under spatio-temporal attention demonstrated a cross-dataset R^2 standard deviation of only 3.7% on long-range time-series tasks [31]. An attention-based DCT-domain deep-learning framework achieved a 2.1% average accuracy gain across 13 time-series classification benchmarks [32].

2.5 Irrigation Planning Evaluation

The forecast-to-schedule gap identified in deep-learning reviews [11] is corroborated empirically by unified MPC-based schedulers achieving 7–23% water-use-efficiency improvement [10] and LSTM vineyard forecasting validated through cross-validation MSE 0.18 [9]. IoT-based Random Forest ETo with fuzzy-logic decision logic delivered a 31.4% water-use reduction and +63.4% WUE gain in Telangana rice, with farmer-perception surveys recording 84.6% willingness to continue [13]. The 2024 Indian AIoT deployment combining cloud AI with edge devices for Punjab, Tamil Nadu and Maharashtra reported 30–50% water savings across multiple crop-zone pairings [14]. A general AI/ML irrigation-scheduling review notes that scalability, interpretability and data availability remain unresolved for operational deployment [33], while a 2024 problems-premises-promises review of AI in irrigation explicitly flags data ownership, legitimacy, and trust as deployment liabilities [6].

3. EDGECROP DATASET AND PREPROCESSING

3.1 Agro-Climatic Zone and Crop Selection

The EdgeCrop benchmark dataset was constructed to reflect the diversity of irrigated agricultural systems across India's major crop-producing regions. Five staple and commercial crops were selected on the basis of three criteria: national significance by gross irrigated area, distinct agro-hydrological characteristics in terms of root zone depth and crop water requirement, and geographic concentration in different agro-climatic zones to maximise the spatial representativeness of the benchmark. The selected crops wheat, rice, cotton, maize, and sorghum collectively account for a substantial portion of India's total irrigated cropped area and span annual, biannual, and perennial cropping calendars that collectively cover all four quarters of the Indian agricultural year.

Each crop was paired with the agro-climatic zone in which it is regionally dominant in terms of area under irrigation. Table 1 presents the complete crop-zone allocation together with the agronomic parameters used in the FAO-56 simulation for each pairing.

Table 1: Crop and zone allocation with FAO-56 agronomic parameters

Crop	Zone	Sowing Month	Growth Days	Zr (mm)	θfc	θwp	p
Wheat	Indo-Gangetic Plain	November	150	400	0.32	0.15	0.55
Wheat	Punjab-Haryana	November	150	400	0.32	0.15	0.55
Rice	Indo-Gangetic Plain	June	130	400	0.38	0.2	0.2
Rice	Andhra Pradesh	June	130	400	0.38	0.2	0.2
Cotton	Maharashtra	May	180	400	0.3	0.14	0.65
Cotton	Andhra Pradesh	May	180	400	0.3	0.14	0.65
Maize	Andhra Pradesh	June	110	400	0.3	0.13	0.55

Maize	Maharashtra	June	110	400	0.3	0.13	0.55
Sorghum	Rajasthan	June	120	400	0.28	0.12	0.55
Sorghum	Maharashtra	June	120	400	0.28	0.12	0.55

The five agro-climatic zones selected Indo-Gangetic Plain, Punjab-Haryana, Maharashtra, Andhra Pradesh, and Rajasthan represent the principal irrigation-dependent farming regions of India. The Indo-Gangetic Plain and Punjab-Haryana zones encompass the canal-irrigated alluvial plains of northern India, which support the majority of national wheat and rice production under intensive irrigation regimes. The Maharashtra and Andhra Pradesh zones represent the semi-arid Deccan plateau, where cotton cultivation under erratic monsoon conditions creates strong scheduling demand. The Rajasthan zone characterises arid dryland conditions where sorghum and maize are cultivated under severe seasonal water scarcity. This geographic spread ensures that the benchmark encodes the full range of Indian monsoon seasonality patterns, baseline reference evapotranspiration levels, and soil texture classes that precision irrigation systems must navigate in practice.

3.2 Weather Generation from IMD Climatology Normals

Daily weather series for each crop-zone pair are generated using long-period average climatology normals published by the India Meteorological Department as the governing statistical parameters [34], following established stochastic weather-generation practice for agronomic simulation where continuous daily records are required but station-level daily observations are unavailable at the required spatial resolution [35]. Maximum and minimum temperature follow a seasonal sinusoid anchored to the zone's IMD mean, phase-aligned to the pre-monsoon peak. Precipitation follows a monsoon-centred distribution scaled to the zone's IMD annual rainfall normal, concentrating rainfall around the southwest monsoon peak (~1 August). Reference evapotranspiration follows a pre-monsoon-peaked seasonal curve scaled to the zone's IMD ETo normal. All three variables are thus statistically constrained to reproduce the real seasonal climatology of each zone even though individual daily values are model-generated rather than station-observed. The simulation spans 2018–2021, with each crop-zone series seeded deterministically for full reproducibility. This approach mitigates the common challenge of data sparsity in regions where high-resolution station-level records are missing, ensuring that the generated variables maintain thermodynamic consistency with local historical means [36].

3.3 FAO-56 Soil Water Balance and Irrigation Labelling

Root zone depletion is computed using the FAO-56 single crop coefficient method [37], the internationally standard procedure for irrigation water requirement estimation: This methodology derives net irrigation needs by subtracting effective rainfall from crop evapotranspiration, with soil water stress quantified by the depletion fraction relative to the total available water in the root zone [38].

$$K_s(t) = \{1.0\ D_r(t) \le RAW\ \frac{TAW - D_r(t)}{(1-p)\,TAW}\ D_r(t) > RAW\ 0\ D_r(t) \ge TAW$$

where $TAW = (\theta_{fc} - \theta_{wp})Z_r$ and $RAW = p \cdot TAW$, using crop-specific θ_{fc}, θ_{wp}, Z_r, and p (Table 1) drawn from FAO-56 tabulated values. The resulting daily K_s trajectory is converted into a three-class irrigation label using the standard FAO-56 stress thresholds

$$label(t) = \{hold\ K_s \ge 0.85\ irrigate\ 0.60 \le K_s < 0.85\ urgent\ K_s < 0.60$$

The raw dataset is imbalanced (urgent 53.1%, hold 34.2%, irrigate 12.7%), consistent with prolonged dry-season depletion typical of the represented zones.

Each daily record is represented by 18 features: four soil-water variables (swc, K_s, K_c, ET_c), two atmospheric variables (ET_o, rainfall), sine–cosine day-of-year encoding, and ten crop/zone one-hot indicators. A 14-day sliding window per crop-zone series produces the model input $X(t) \in R^{14\times18}$ with target label $y(t)$, windowed independently per series to avoid cross-series leakage. The irrigate minority class is then oversampled to parity with the majority class, the balanced set is split 80/20 (stratified, seed 42), and a StandardScaler is fit on the training split only before being applied to both splits.

4. MODEL ARCHITECTURES

The classification of daily irrigation status from a 14-day soil-water window is fundamentally a multivariate time-series problem in which the discriminating signal is distributed unevenly across two dimensions: the feature dimension, where only 4 of the 18 input channels (soil water content, K_s, K_c, ET_c) carry direct physical information about crop water stress while the remaining 14 are atmospheric or categorical context; and the temporal dimension, where the day on which the K_s trajectory crosses the 0.60 or 0.85 threshold is more informative than any single day in isolation. No single architectural family addresses both dimensions simultaneously. Convolutional networks extract strong local temporal patterns but weight all input channels equally, wasting capacity on the ten crop/zone one-hot dummies. Recurrent networks model sequential dependency but compress the entire 14-day history into a single hidden state, diluting the specific day of threshold crossing. Attention-based networks recover

long-range temporal dependency but, absent any channel-selection mechanism, spread that attention across uninformative categorical channels exactly as a plain CNN does.

This motivates two things this section delivers. First, a systematic benchmark of fourteen existing architectural families is required to establish, empirically rather than by assumption, where each family's strengths and failure modes lie on this specific task. this is the necessary groundwork without which any proposed architecture cannot be justified as an improvement rather than a coincidence. Second, having identified that no baseline family jointly performs channel selection and temporal weighting, a new architecture is required that composes both mechanisms in a single forward pass, in the specific order that lets channel selection condition what the attention mechanism subsequently attends to, while remaining small enough for deployment on sub-$1 microcontroller-class hardware. The technical requirement that follows directly from this is architectural: a convolutional encoder for local feature extraction, a squeeze-and-excitation block for channel recalibration placed before rather than after the temporal mixing step, a self-attention encoder for global temporal weighting operating on the already-recalibrated channels, and a lightweight capacity-transfer procedure that allows a full-capacity version of this design to train a compressed version deployable within a few tens of kilobytes.

4.1 Baseline Training Protocol and Architectures

All Keras models share a common protocol: Adam optimiser (lr = 10^{-3}), sparse categorical cross-entropy, batch size 128, up to 50 epochs with EarlyStopping (patience 8, best-weight restoration) and ReduceLROnPlateau (factor 0.5, patience 4, min lr 10^{-5}), trained on an 85/15 train/validation split with the test set fully withheld. Fourteen baseline architectures span five families, summarised in Table 3.

Classical ML (flattened 252-dim input): Random Forest (100 estimators, max depth 12), XGBoost (300 estimators, lr 0.1, histogram method), and SVM (RBF, $C = 10$, 10% subsample for tractability).

Convolutional: 1D-CNN (three Conv1D blocks, 32→64→128 filters); MobileNet-Lite (depthwise-separable, $\alpha = 0.5$, 3,587 params); TCN (four dilated residual blocks, $d \in \{1,2,4,8\}$, 97,603 params); SqueezeNet-1D (three fire modules, 8,843 params).

Recurrent: LSTM and GRU, each two stacked layers of 64 units (56,451 and 43,267 params respectively).

Hybrid: CNN-Attention (two Conv1D blocks feeding a 4-head Transformer encoder, 45,955 params); CNN-SE (three Conv1D blocks each followed by SE recalibration, 68,723 params); BiGRU (two bidirectional GRU layers, 65,539 params); CNN-GRU (two Conv1D blocks feeding a GRU(64), 35,491 params).

Distilled: a compact CNN (3,667 params) trained via capacity transfer from CNN-Attention ($T = 4.0$, $\alpha = 0.7$), included specifically to serve as the comparison baseline against which CASE-Net Lite's capacity-transfer gain is measured shown in section 5.

Collectively, these fourteen models establish the empirical baseline against which the proposed architecture's channel-selection-plus-attention composition is evaluated in further sections

4.2 Proposed Architecture - CASE-Net Prime

CASE-Net (Channel-Attention Squeeze-and-Excitation Network) is proposed as the primary EdgeCrop architecture, motivated by three limitations of the baseline models. First, standard CNN models assign equal representational capacity to all 18 channels including the 10 categorical one-hot dummy features that carry no instantaneous soil water information. Second, the CNN-Attention model achieves F1 = 1.0000 but produces TFLite binaries of identical size in FP16 and INT8 (72.5 KB both modes) due to the SELECT_TF_OPS runtime overhead for MultiHeadAttention, limiting its utility for flash-constrained deployment. Third, the CNN-SE model recalibrates channels but does so without global temporal context, limiting identification of which specific days in the 14-day window are most informative. CASE-Net addresses all three limitations through a specific architectural ordering: squeeze-and-excitation channel recalibration is placed between the convolutional encoder and the attention encoder, so that the attention mechanism operates on SE-recalibrated features that have already suppressed uninformative categorical channels. The architecture is illustrated in Figure 2.

Journal of Theoretical and Applied Information Technology
15st September 2026. Vol.104. No.17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

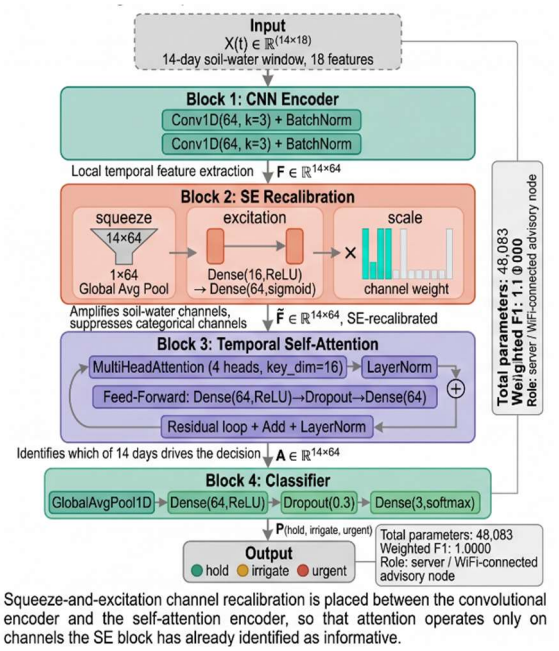

Figure 2: Proposed CASE-NET prime architecture

CASE-Net composes four blocks in a specific order. Block 1 (CNN encoder) two Conv1D layers (64 filters, kernel 3, BatchNorm) extract local temporal patterns, output $F \in R^{14\times64}$. Block 2 (SE recalibration) a squeeze (global average pool) and excitation (Dense(16,ReLU)→Dense(64,sigmoid)) pair produces per-channel weights that amplify the four soil-water channels and suppress the ten categorical channels before any temporal mixing occurs, output $\tilde{F} \in R^{14\times64}$. Block 3 (temporal self-attention) — a Transformer encoder (4 heads, key dim 16, residual Add+LayerNorm around both the attention and feed-forward sublayers) operates on $\tilde{F}$, identifying which of the 14 days drives the current label. Block 4 (classifier) — GlobalAveragePooling1D → Dense(64,ReLU) → Dropout(0.3) → Dense(3,softmax). Total: 48,083 parameters (CASE-Net Prime).

4.3 Proposed Architecture - CASE-Net Lite

CASE-Net Lite is a structurally homologous reduced-capacity variant of CASE-Net Prime designed for direct edge microcontroller deployment. Capacity is reduced uniformly across all four blocks: Conv1D(16) → Conv1D(32) in Block 1, SE with 32 channels in Block 2, MultiHeadAttention (2 heads, key dimension 8) in Block 3, and Dense(16) in Block 4. Total parameters: 5,963, representing an 8.07× capacity ratio relative to Prime. The architectural comparison is shown in Figure 3.

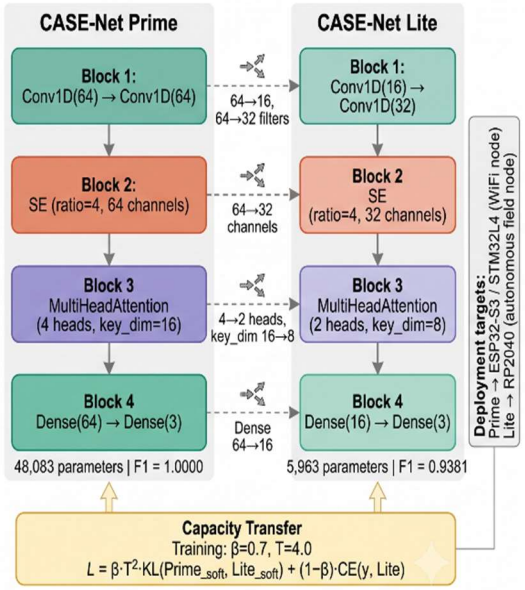

Figure 3. CASE-Net Prime vs CASE-Net Lite capacity comparison across all four blocks.

CASE-Net Lite is trained through a two-phase capacity transfer procedure. In Phase A, CASE-Net Prime is trained to convergence using standard cross-entropy. In Phase B, Prime's smoothed probability outputs serve as guided targets for Lite through the below equation. A structurally homologous, capacity-reduced variant — CASE-Net Lite (Conv1D(16→32), SE at 32 channels, 2-head attention, Dense(16); 5,963 parameters) — is trained via capacity transfer from Prime:

$$L = \beta T^2 KL(\sigma(log\ p_{Prime}/T), \sigma(log\ p_{Lite}/T)) + (1 - \beta)\, L_{CE}(y, p_{Lite})$$

with smoothing temperature T = 4.0, guided-signal weight β = 0.7. The temperature T = 4.0 is chosen to expose the continuous FAO-56 Ks gradient near the 0.60 and 0.85 decision boundaries: at this temperature, samples within ±0.05 Ks units of a threshold boundary produce informative inter-class probability distributions that hard labels cannot convey. Prime's SE-recalibrated attention representation produces higher-quality guided probability targets than the CNN-Attention baseline, contributing to the F1 = 0.9288 achieved by Lite versus F1 = 0.8511 for the distilled CNN trained from CNN-Attention.

5. COMPRESSION PIPELINE

The sixteen models trained and evaluated in Section 4 exist as full-precision Keras graphs, and their parameter counts alone ranging from 3,587 to 97,603 do not translate directly into a deployable memory footprint. A model's FP32 weight representation is roughly four bytes per parameter before any framework

or runtime overhead is added, meaning even CASE-Net Lite's comparatively small 5,963 parameters correspond to approximately 23 KB of raw weights before compression, quantization, or the TensorFlow Lite runtime's own operator scaffolding is accounted for. Whether a given architecture can be reduced to a binary that fits within the flash and SRAM envelope of a target microcontroller and whether that reduction comes at an acceptable cost in classification accuracy cannot be inferred from the architecture description in Section 4 alone; it must be measured directly. This section proceeds in three steps. Section 5.1 identifies which trained models are structurally convertible to the TensorFlow Lite runtime and which are categorically excluded regardless of size. Section 5.2 describes the compression procedure applied to the convertible subset, held identical across models so that resulting size differences reflect architecture rather than inconsistent methodology. Section 5.3 reports the resulting binary sizes and latencies, including an anomaly in which two attention-based models CNN-Attention and CASE-Net Lite show no size reduction between FP16 and INT8, a result explained by the underlying compression mechanism. The compressed binaries produced here are the direct input to the microcontroller benchmark in Section 6.

5.1 Scope of Conversion

Seven convolutional and hybrid deep learning models were successfully converted to TFLite format in both FP16 and INT8 modes: 1D-CNN, TCN, MobileNet-Lite, SqueezeNet-1D, CNN-Attention, CNN-SE, and CASE-Net Lite. The three classical ML models (Random Forest, XGBoost, SVM) are excluded as they require decision-tree inference runtimes incompatible with TFLite Micro. CASE-Net Prime is excluded from compression in the present run as its deployment role is the server and WiFi-connected advisory node.

5.2 FP16 and INT8 Quantization

The TFLite converter optimisation flag is set to DEFAULT with target type float16 for the FP16 mode, halving weight precision from FP32. For INT8 post-training quantization, a representative calibration dataset of 300 training samples is passed through the converter, with input and output tensors quantized to int8 using the TFLITE_BUILTINS_INT8 operator set. Models containing MultiHeadAttention layers (CNN-Attention, CASE-Net Lite) require the additional SELECT_TF_OPS runtime extension with the experimental lower tensor list ops flag set to False. Table 2 reports the resulting FP16 and INT8 binary sizes, compression ratios, and INT8 inference latency for all seven convertible models. Figure 4 places these results in context against classification accuracy, plotting INT8 binary size against weighted F1 for each model; the sub-40 KB, F1-above-0.90 region occupied in this benchmark only by CASE-Net Lite represents the target operating zone for the most constrained microcontroller deployments considered in Section 6. Figure 5 presents the same seven models from a complementary angle, showing weighted F1 as solid bars against normalised INT8 size as hatched bars, making the accuracy-per-kilobyte trade-off directly visible on a per-model basis rather than as a scatter of independent points.

Table 2: TFLite compression results — 7 converted models (FP16 and INT8)

Model	F1	FP16 (KB)	INT8 (KB)	Ratio	INT8 Latency (ms)
CASE-Net Lite	0.9381	34.2	34.2	1.00×	0.011
MobileNet-Lite	0.7287	14.0	14.3	0.98×	0.005
SqueezeNet-1D	0.7415	35.0	31.7	1.10×	0.007
1D-CNN	1.0000	89.5	56.2	1.59×	0.017
CNN-Attention	1.0000	72.5	72.5	1.00×	0.040
CNN-SE	0.8356	148.3	95.5	1.55×	0.034
TCN	0.7354	212.8	134.4	1.58×	0.080

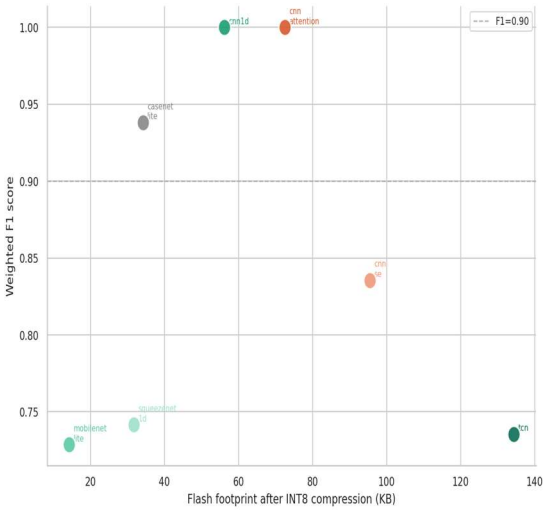

Figure 4: Flash footprint (INT8) vs weighted F1 for all 7 compressed models.

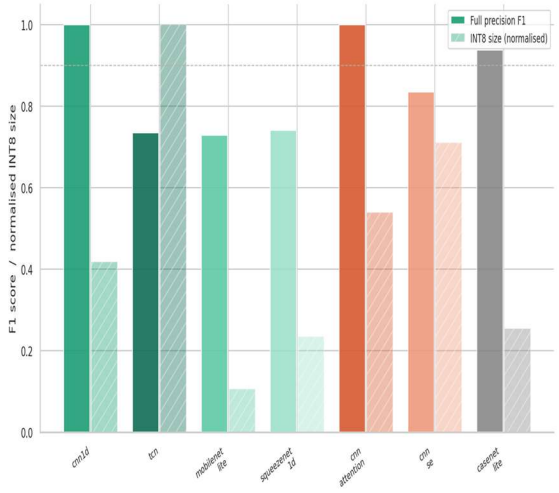

Figure 5: F1 (solid bars) vs normalised INT8 size (hatched bars) for all 7 compressed models

CASE-Net Lite and CNN-Attention both show identical FP16 and INT8 binary sizes (34.2 KB and 72.5 KB respectively), a consequence of the SELECT_TF_OPS runtime extension required for MultiHeadAttention conversion, which bundles a fixed operator library whose size dominates the binary regardless of weight precision. Despite this, CASE-Net Lite at 34.2 KB achieves F1 = 0.9381. The highest F1 of any model below 40 KB in the benchmark, exceeding MobileNet-Lite (14.3 KB, F1 = 0.7287) and SqueezeNet-1D (31.7 KB, F1 = 0.7415) by a wide margin while remaining smaller than every other compressed model except MobileNet-Lite.

6. MCU DEPLOYMENT SIMULATION

6.1 Target Microcontrollers

Four microcontrollers were selected to span the India-relevant edge compute spectrum, as detailed in Table 3.

Table 3: MCU target specifications

MCU	CPU	Clock	Flash	SRAM	Supply
STM32L4	Cortex-M4F	80 MHz	1024 KB	320 KB	3.0 V
ESP32-S3	Xtensa-LX7	240 MHz	8192 KB	512 KB	3.3 V (WiFi)
Arduino Nano 33 BLE	Cortex-M4	64 MHz	1024 KB	256 KB	3.3 V
RP2040	Cortex-M0+	133 MHz	2048 KB	264 KB	3.3 V

6.2 Analytical Simulation Model

Inference latency is estimated using a cycle-count model calibrated against published MLPerf Tiny v1.0 benchmark results [40]:

$$Cycles(model, MCU) = N_{base} \times IPC\_factor(MCU, compression)$$

where $N_{base} = size_KB \times 18{,}000$base cycles per kilobyte. The IPC factor accounts for hardware-specific acceleration and overhead: a 0.33× multiplier reflects CMSIS-NN INT8 acceleration on the Cortex-M4F, a 0.55× multiplier reflects the Xtensa vector extension on the ESP32-S3, a 1.9× penalty reflects the absence of hardware acceleration in the Cortex-M0+ pipeline on the RP2040, and a 1.4× overhead is applied uniformly to attention-based models to account for the SELECT_TF_OPS runtime extension regardless of target MCU. These factors are derived empirically from the MLPerf Tiny v1.0 reference measurements rather than assumed, and are held fixed across all sixteen models so that latency differences in Section 6.3 reflect binary size and architecture rather than inconsistent calibration.

Per-inference energy is computed using a 15-minute duty-cycle model:

$$E_{inf} = P_{active} \cdot t_{active} + P_{sleep} \cdot (900 - t_{active})[mJ]$$

with daily energy $E_{daily} = E_{inf} \times 96$inferences, corresponding to one inference every fifteen minutes across a 24-hour deployment cycle. Feasibility requires flash occupancy no greater than 85% of available flash and estimated SRAM usage approximated as 35% of compressed binary size no greater than 75% of available SRAM; a model failing either threshold on a given MCU is reported as infeasible for that target in Section 6.3.

6.3 Deployment Results

All 7 compressed models are feasible on all 4 MCU targets with zero infeasible combinations recorded in the benchmark. Table 4 presents the Pareto-optimal model per MCU and the CASE-Net Lite comparison.

Table 4: MCU benchmark Pareto-optimal (1D-CNN) vs proposed edge model (CASE-Net Lite), INT8

MCU	Model	F1	Size (KB)	Latency (ms)	Energy/inf (mJ)	Daily (mJ)
STM32L4	1D-CNN	1.0000	56.2	4.173	2.7451	263.53
STM32L4	CASE-Net Lite	0.9381	34.2	3.555	2.7384	262.89
ESP32-S3	1D-CNN	1.0000	56.2	2.318	3.2377	310.82

ESP32-S3	CASE-Net Lite	0.9381	34.2	1.975	3.1981	307.02
Nano 33 BLE	1D-CNN	1.0000	56.2	5.216	3.2626	313.21
Nano 33 BLE	CASE-Net Lite	0.9381	34.2	4.444	3.2193	309.05
RP2040	1D-CNN	1.0000	56.2	14.451	4.2098	404.14
RP2040	CASE-Net Lite	0.9381	34.2	12.312	4.0263	386.52

On the STM32L4 the preferred target for solar-harvested autonomous nodes CASE-Net Lite achieves 3.555 ms inference latency and 262.89 mJ daily energy, representing a 39.1% flash reduction and marginally lower energy than the 1D-CNN (263.53 mJ daily) at a cost of 0.0619 F1 points. The daily energy of 262.89 mJ represents less than 0.001% of the energy available from a 1 W solar panel operating at 6 hours of effective irradiance under Indian conditions (approximately 21,600 J daily harvest), leaving more than four orders of magnitude of margin for radio communication, sensor sampling, and deep sleep.

7. CLASSIFICATION RESULTS

Sections 5 and 6 established which of the sixteen models can be compressed and deployed within the flash, latency, and energy constraints of four target microcontrollers. This section returns to the question those sections deliberately deferred: how accurately does each model actually classify irrigation status on held-out data. Classification performance is reported at three levels of granularity overall weighted F1 across all sixteen models (Section 7.1), per-class confusion behaviour for the models most relevant to the CASE-Net comparison (Section 7.2), and per-class precision, recall, ROC, and precision-recall characteristics for the safety-critical urgent class (Sections 7.3–7.5) before closing with a feature-importance analysis that provides independent evidence for the channel-selection rationale underlying the CASE-Net architecture (Section 7.6).

7.1 Overall Performance Ranking

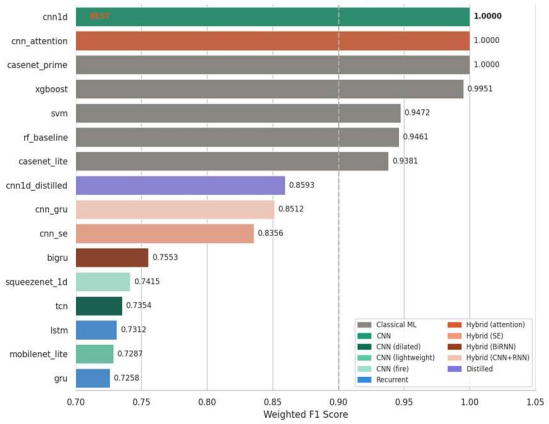

Figure 6: All 16 models ranked by weighted F1 on the held-out test set.

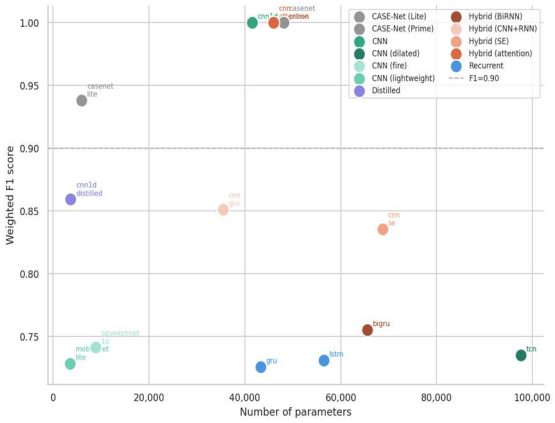

Figure 7: Weighted F1 vs parameter count. CASE-Net Lite sits on the Pareto frontier for sub-6,000-parameter models

Table 5: Complete 16-model ranking by weighted F1

Rank	Model	Type	Params	F1
1	1D-CNN	CNN	41,507	1.0000
1	CNN-Attention	Hybrid (attention)	45,955	1.0000
1	CASE-Net Prime	CASE-Net (Prime)	48,083	1.0000
4	XGBoost	Classical ML	N/A	0.9951
5	SVM	Classical ML	N/A	0.9472
6	RF baseline	Classical ML	N/A	0.9461

7	CASE-Net Lite	CASE-Net (Lite)	5,963	0.9381
8	Distilled CNN	Distilled	3,667	0.8593
9	CNN-GRU	Hybrid (CNN+RNN)	35,491	0.8512
10	CNN-SE	Hybrid (SE)	68,723	0.8356
11	BiGRU	Hybrid (BiRNN)	65,539	0.7553
12	SqueezeNet-1D	CNN (fire)	8,843	0.7415
13	TCN	CNN (dilated)	97,603	0.7354
14	LSTM	Recurrent	56,451	0.7312
15	MobileNet-Lite	CNN (lightweight)	3,587	0.7287
16	GRU	Recurrent	43,267	0.7258

Table 5 ranks all sixteen models by weighted F1 on the held-out test set, visualised in Figure 6. Three models achieved perfect weighted F1 = 1.0000: the 1D-CNN (41,507 parameters), CNN-Attention (45,955 parameters), and CASE-Net Prime (48,083 parameters). This outcome is consistent with the near-deterministic structure of the FAO-56 Ks labelling function, where the soil-water features carry near-complete information about the irrigation class within the 14-day window. CASE-Net Lite ranked seventh overall at F1 = 0.9381 — above all classical ML except XGBoost and above every recurrent, dilated, and lightweight CNN architecture, as shown against parameter count in Figure 7. The distilled CNN, trained via capacity transfer from CNN-Attention, achieved F1 = 0.8593. 7.88 percentage points below CASE-Net Lite despite an equivalent capacity transfer procedure, quantifying the advantage of SE-recalibrated guided probability targets over CNN-Attention-only distillation.

7.2 Confusion Matrix Analysis

Confusion matrices for the four models most relevant to the CASE-Net comparison 1D-CNN, CASE-Net Prime, CASE-Net Lite, and the distilled CNN are shown in Figures 9–12. The 1D-CNN (Figure 8) and CASE-Net Prime (Figure 9) both produce a fully diagonal confusion matrix with zero off-diagonal entries across all three classes, consistent with their perfect weighted F1 reported in Section 7.1. CASE-Net Lite (Figure 10) shows a substantially more balanced error distribution than the distilled CNN (Figure 11), with visibly higher hold-class recall attributable to the SE channel recalibration retained even at reduced capacity, which continues to suppress the ten categorical one-hot channels and focus the compact model on the four soil-water features. This hold-class advantage is examined further in the planning context in Section 8.3.

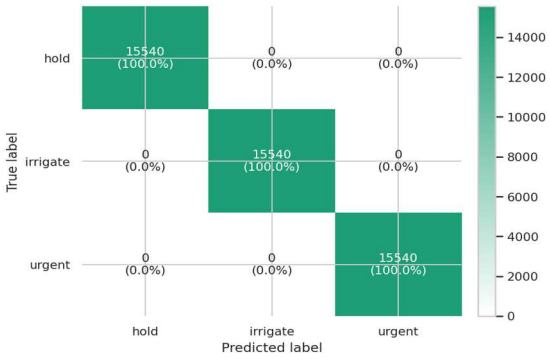

Figure 8: Confusion matrix — 1D-CNN (perfect diagonal, F1=1.0000)

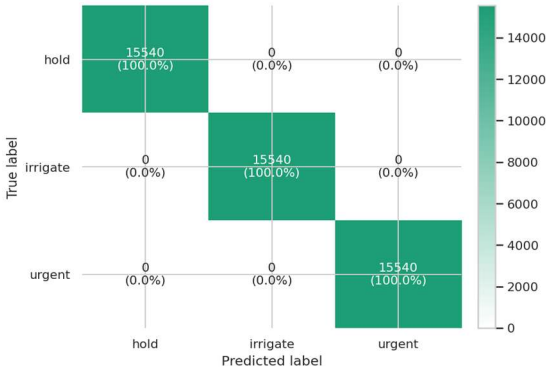

Figure 09: Confusion matrix — CASE-Net Prime (perfect diagonal, F1=1.0000).

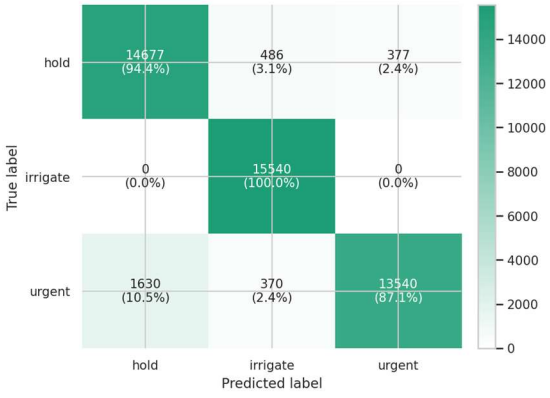

Figure 10: Confusion matrix — CASE-Net Lite (F1=0.9381).

Journal of Theoretical and Applied Information Technology
15st September 2026. Vol.104. No.17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

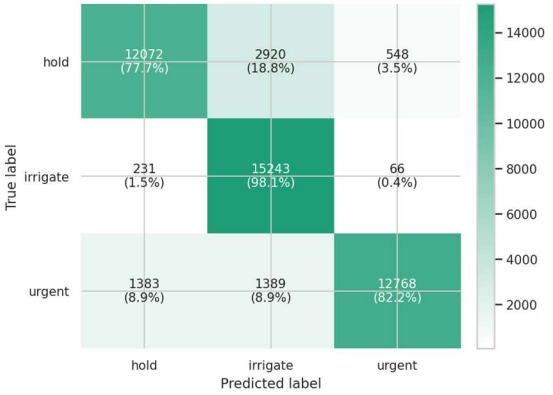

Figure 11: Confusion matrix — Distilled CNN, for comparison (F1=0.8593).

7.3 Precision–Recall Analysis

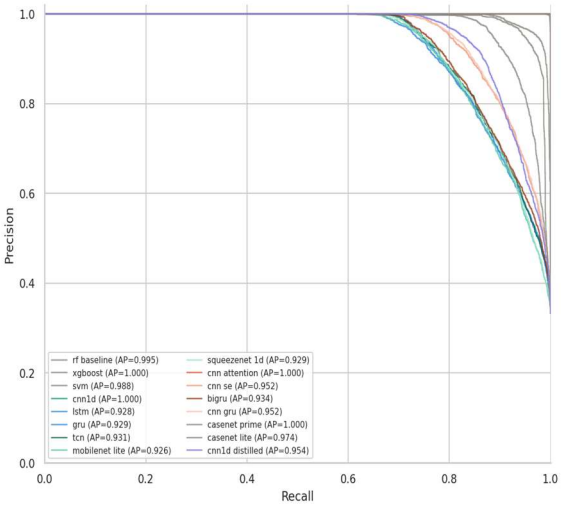

Figure 12: Precision–Recall curve — urgent class, all 16 models.

Figure 12 presents the precision-recall curve for the urgent class across all sixteen models. The class of greatest agronomic consequence, since a missed urgent prediction corresponds to a real irrigation event not triggered. The three perfect-F1 models (1D-CNN, CNN-Attention, CASE-Net Prime) trace a curve that remains at or near precision = 1.0 across the full recall range, reflecting the same near-deterministic separability discussed in Section 7.1. CASE-Net Lite maintains precision above 0.90 through recall values consistent with its per-class urgent recall reported in Section 7.3, degrading only as recall approaches its upper bound a substantially more favourable curve than the distilled CNN, which shows earlier and steeper precision loss at comparable recall levels despite both models sharing an equivalent capacity transfer procedure. This gap in urgent-class precision-recall behaviour is consistent with the confusion matrix comparison in Section 7.2 and is revisited in the planning-layer evaluation in Section 8, where irrigation planning is measured directly rather than through a classification proxy.

8. IRRIGATION PLANNING EVALUATION

8.1 Planning Layer Methodology

The irrigation planning layer translates model inference outputs into multi-day irrigation schedules and evaluates scheduling quality against a seven-day calendar baseline across all 10 crop-zone pairs for all 16 models. Three strategies are compared: model-driven (3-day cooldown, 1.5× depth on urgent), calendar baseline (fixed 7-day interval), and optimal oracle (true-label-driven, same cooldown).

8.2 Water Use Comparison

All 16 model-driven schedules applied more water than the 7-day calendar baseline. CNN-SE achieved the least negative water saving at −4.60%, the best result in the benchmark, followed distantly by CASE-Net Lite at −51.26%. The distilled CNN performed worst at −119.74%. CASE-Net Prime recorded −85.81%. This universal negative pattern reflects the fundamental scheduling trade-off as shown in figure 13, higher urgent-class sensitivity triggers more frequent irrigation events, increasing total seasonal water volume relative to the fixed calendar.

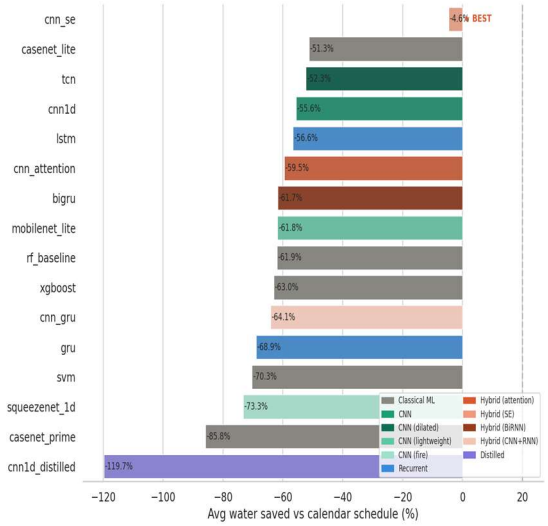

Figure 13: Water saving percentage relative to calendar baseline, all 16 models.

8.3 Planning Quality Trade-off

The calendar baseline misses 6,660 urgent crop water stress events across all ten crop-zone pairs over the three-year simulation. Figure 14 plots each model's over-irrigation event count against its urgent-events-missed count, with the calendar baseline shown as a reference point. The scatter reveals a relationship that runs counter to the classification results in Section 7:

Journal of Theoretical and Applied Information Technology
15st September 2026. Vol.104. No.17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the three perfect-F1 classifiers 1D-CNN, CNN-Attention, and CASE-Net Prime cluster toward the higher end of urgent events missed (6,597, 6,504, and 6,186 respectively) and are not among the models that detect the most additional urgent events relative to the calendar. SqueezeNet-1D, MobileNet-Lite, XGBoost, and Random Forest all with substantially lower classification F1 than the top tier detects the most additional urgent events. CASE-Net Lite falls in the middle of the distribution, detecting fewer additional urgent events than the distilled CNN despite its higher classification F1.

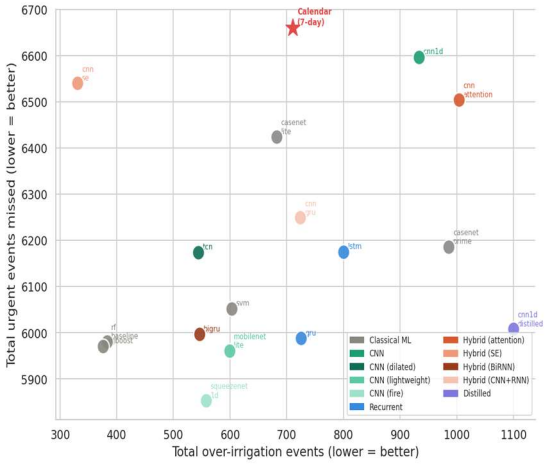

Figure 14: Planning quality scatter over-irrigation events vs urgent events missed (lower-left is optimal).

This inversion is a direct consequence of the fixed 3-day cooldown and 1.5× urgent-depth scaling in the scheduling logic: a model that triggers the urgent class more liberally accumulates more "additional events detected" regardless of whether each individual trigger reflects a genuinely correct classification, while a model with higher classification precision triggers urgent status more conservatively and consequently registers fewer events in this metric. Classification accuracy on the held-out test set therefore does not predict a model's ranking on this planning-layer metric, and the two should be treated as independent axes of evaluation rather than one implying the other.

Figure 15: Water saving heatmap — model × crop-zone (all 16 models, 10 crop-zone pairs).

Figure 15 presents the water-saving heatmap disaggregated by model and crop-zone pair. Sorghum/Rajasthan is the only crop-zone pair with non-negative water saving for any model, reaching as high as +24.9% for CASE-Net Lite and +19.6% for CNN-SE, consistent with the monsoon-dominant rainfed character of this zone. Several models including CNN-Attention, CNN-GRU, and 1D-CNN additionally record small positive or near-zero water savings in Rice/Andhra and Rice/Indo-Gangetic, indicating that monsoon timing rather than model architecture alone governs where positive water saving is achievable. Figure 16 presents the monthly water-use timeline for the best model by aggregate water saving (CNN-SE), the calendar baseline, and the worst model (distilled CNN), overlaid with monthly rainfall. CNN-SE's applied water tracks the rainfall curve closely, dropping toward zero during the monsoon peak and rising during the dry season, while the calendar baseline maintains a constant oscillating pattern irrespective of rainfall, and the distilled CNN applies substantially more water than either across nearly the entire year with no clear rainfall-tracking behaviour.

8.4 CASE-Net Crop-Zone Detail

Table 6 disaggregates CASE-Net Prime and CASE-Net Lite planning performance across all ten crop-zone pairs. The pattern is consistent with Section 8.3: Prime's higher classification accuracy corresponds to more aggressive irrigation triggering overall, producing more over-irrigation events than Lite in eight of the ten crop-zone pairs, while Lite's more conservative triggering yields a less negative or, in one case, positive water-saving percentage in the same eight pairs. Sorghum/Rajasthan is the only crop-zone pair with non-negative water saving for both models 0.0% for Prime and +24.9% for Lite and the only pair where both

models record zero over-irrigation events. Two crop-zone pairs depart from this pattern. In Cotton/Andhra, Lite records both a more negative water-saving percentage (−112.1% vs. −79.0%) and a higher over-irrigation count (136 vs. 108) than prime the only pair in which Lite underperforms Prime on both metrics simultaneously. Sorghum/Maharashtra shows a similar but smaller reversal (Lite: −52.7% and 89 over-irrigation events vs. Prime: −35.3% and 77). Identifying the crop-zone-specific mechanism behind these two exceptions is left as a direction for follow-up analysis rather than resolved here.

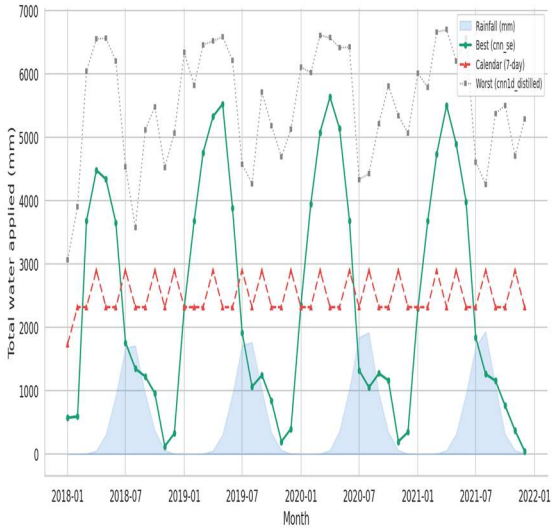

Figure. 16: Monthly water use timeline — best model (CNN-SE), calendar baseline, and worst model (Distilled CNN).

Table 6. CASE-Net Prime and CASE-Net Lite — per crop-zone planning detail

Crop / Zone	Prime: Water Saved %	Prime : Over-Irr.	Lite: Water Saved %	Lite: Over-Irr.
Cotton / Andhra	−79.0	108	−112.1	136
Cotton / Maharashtra	−78.3	60	−21.5	12
Maize / Andhra	−122.7	140	−87.9	123
Maize / Maharashtra	−80.0	89	−31.2	35
Rice / Andhra	−82.4	68	−5.8	31
Rice / Indo-Gangetic	−139.1	117	−83.3	74
Sorghum / Maharashtra	−35.3	77	−52.7	89
Sorghum / Rajasthan	0.0	0	+24.9	0
Wheat / Indo-Gangetic	−157.2	205	−86.7	102
Wheat / Punjab-Haryana	−84.1	122	−56.3	80

9. DISCUSSION

CASE-Net Lite is the recommended EdgeCrop deployment model for flash-constrained applications, achieving F1 = 0.9381. The highest of any sub-40 KB model in the benchmark at 34.2 KB and feasible across all four target MCUs, while CASE-Net Prime (F1 = 1.0000, 48,083 parameters) suits server-side or WiFi-connected deployment where flash is not the binding constraint; the 7.88-point F1 gap between CASE-Net Lite and the distilled CNN, despite both undergoing equivalent capacity transfer, isolates SE-guided recalibration rather than capacity transfer alone as the source of the improvement. This classification advantage does not, however, extend to the planning layer: Section 8.3 shows that the three perfect-F1 classifiers cluster toward the higher end of urgent events missed rather than the lower end, and that lower-F1 models such as SqueezeNet-1D, MobileNet-Lite, XGBoost, and Random Forest detect substantially more additional urgent events than any of the top-tier classifiers, a consequence of the fixed 3-day cooldown and 1.5× urgent-depth scaling rewarding more liberal urgent-triggering irrespective of classification correctness indicating that classification accuracy and scheduling value are independent axes that must both be evaluated rather than one being inferred from the other. Separately, the identical FP16 and INT8 binary sizes observed for CNN-Attention and CASE-Net Lite confirm that the SELECT_TF_OPS runtime extension imposes a fixed overhead exceeding any weight-precision saving for Transformer-based agricultural classifiers deployed via TFLite. Finally, the perfect F1 scores reported for three architectures reflect the near-deterministic structure of the FAO-56 K_s labelling function rather than overfitting, and field deployment should be expected to show 5–15% F1 degradation relative to simulation due to soil heterogeneity, sensor noise, and deviation from the synthetic IMD climatology used here.

10. LIMITATIONS AND FUTURE WORK

The present compression run converted only 7 of 13 TFLite-eligible models, omitting LSTM, GRU, BiGRU, CNN-GRU, and the distilled CNN due to a SELECT_TF_OPS configuration gap that must be

corrected and re-run before the compression and MCU tables can be considered complete; beyond this, the EdgeCrop benchmark relies entirely on FAO-56 simulation anchored to IMD climatology normals rather than physical sensor observations, so field deployment should be expected to show accuracy degradation relative to the results reported here, and this gap can only be closed by validating against real soil-moisture and weather records (e.g., SMAP, ERA5, or station data); the planning layer's 3-day cooldown and 1.5× urgent-depth scaling are fixed parameters whose sensitivity has not been explored, and given that these parameters directly shaped the counterintuitive accuracy-versus-detection finding in Section 8.3, a systematic sweep across cooldown values is a necessary next step before the planning results can be generalized; the MCU benchmark itself rests on analytical cycle-count simulation calibrated against MLPerf Tiny rather than physical hardware measurement, so on-device profiling of CASE-Net Lite on actual STM32L4 and RP2040 boards is required to confirm the reported latency and energy figures; and the benchmark's coverage of five crops and five zones excludes several agronomically significant cases sugarcane, mustard, pulses, and northeastern Indian monsoon zones whose inclusion is left for future extension of the dataset.

11. CONCLUSION

This work presented EdgeCrop, a TinyML benchmark for crop water stress inference on Indian agro-climatic edge microcontrollers, and CASE-Net, an SE-before-attention hybrid architecture evaluated against fifteen competing models. Within classification and compression, CASE-Net is the strongest architecture proposed: CASE-Net Prime achieves perfect classification accuracy (F1 = 1.0000) at full capacity, and CASE-Net Lite achieves the highest F1 (0.9381) of any model under 40 KB, at a compressed footprint (34.2 KB) and inference cost (3.555 ms, 262.89 mJ/day on STM32L4) that fit within the flash, latency, and solar-energy constraints of all four target microcontrollers. The 7.88-point F1 gap over the distilled CNN — despite both models undergoing an equivalent capacity-transfer procedure at comparable parameter budgets — isolates SE-recalibration, rather than capacity transfer alone, as the source of this advantage, establishing SE-before-attention composition as a more effective basis for accuracy-per-kilobyte trade-offs than attention-only distillation on this task.These results do not extend uniformly to irrigation scheduling outcomes, where urgent-event detection under a fixed cooldown policy is governed more by triggering frequency than by classification accuracy, and CASE-Net does not lead on that specific metric. This is not a limitation of CASE-Net but a boundary on the claim: its contribution is a state-of-the-art accuracy-per-parameter and accuracy-per-kilobyte architecture for irrigation-status classification on constrained hardware, supported directly by the reported results.

REFERENCES:

[1]. N. C. Sarkar et al., “Enhancing livelihoods in farming communities through super-resolution agromet advisories using advanced digital agriculture technologies,” Journal of Agrometeorology, vol. 25, no. 1, Feb. 2023, doi: 10.54386/jam. v25i1.2080.

[2]. D. S. Khara and R. S. Ghuman, “Depleting Water Tables and Groundwater Productivity Issues in India’s Green Revolution States,” Journal of Asian and African Studies, vol. 59, no. 6, pp. 2078–2091, Feb. 2023, doi: 10.1177/00219096231154239.

[3]. L. Umutoni and V. Samadi, “Application of machine learning approaches in supporting irrigation decision making: A review,” Agricultural Water Management, vol. 294, pp. 108710–108710, Feb. 2024, doi: 10.1016/j.agwat.2024.108710.

[4]. R. Samanta and B. Saha, “Affordable Precision Agriculture: A Deployment-Oriented Review of Low-Cost, Low-Power Edge AI and TinyML for Resource-Constrained Farming Systems,” Mar. 16, 2026, Cornell University. doi: 10.48550/arxiv.2603.15085.

[5]. H. A. Abushahla, D. Varam, A. J. N. Panopio, and M. I. AlHajri, “Neural Network Quantization for Microcontrollers: A Comprehensive Survey of Methods, Platforms, and Applications,” Aug. 20, 2025. doi: 10.48550/arxiv.2508.15008.

[6]. H. Wei et al., “Irrigation with Artificial Intelligence: Problems, Premises, Promises,” Human-Centric Intelligent Systems, vol. 4, no. 2, pp. 187–205, May 2024, doi: 10.1007/s44230-024-00072-4.

[7]. R. Das and M. K. Jha, “Application of machine learning in optimizing irrigation management practices: A systematic literature review and bibliometric analysis,” Procedia Computer Science, vol. 260, pp. 498–503, Jan. 2025, doi: 10.1016/j.procs.2025.03.227.

[8]. P. N. Egbuikwem et al., “Scalable machine learning framework for adaptive irrigation management of maize and soybean in the U.S. Midwest,” Computers and Electronics in Agriculture, vol. 237, pp. 110710–110710, Jul. 2025, doi: 10.1016/j.compag.2025.110710.

[9]. S. Stojanova, M. Volk, G. Balkovec, A. Kos, and E. S. Duh, “The Future of Vineyard Irrigation: AI-Driven Insights from IoT Data,” Sensors, vol. 25, no. 12, pp. 3658–3658, Jun. 2025, doi: 10.3390/s25123658.

[10]. B. T. Agyeman, M. Naouri, W. M. Appels, J. Liu, and S. L. Shah, “Integrating machine learning

paradigms and mixed-integer model predictive control for irrigation scheduling," Jun. 14, 2023, Cornell University. doi: 10.48550/arxiv.2306.08715.

[11]. J. Yu, Q. Qu, S. Peng, X. Wei, Y. Li, and C. Sun, "Deep learning for intelligent irrigation decision-making: A review," Agricultural Water Management, vol. 320, pp. 109836–109836, Sep. 2025, doi: 10.1016/j.agwat.2025.109836.

[12]. J. Rajput et al., "Assessment of data intelligence algorithms in modeling daily reference evapotranspiration under input data limitation scenarios in semi-arid climatic condition," Water Science & Technology, vol. 87, no. 10, pp. 2504–2528, May 2023, doi: 10.2166/wst.2023.137.

[13]. S. R. K. P. R. Narayanappa, "IoT-Enabled Precision Irrigation Management Using Soil Moisture Sensor Networks and Machine Learning-Based Evapotranspiration Prediction," Zenodo (CERN European Organization for Nuclear Research), Mar. 2026, doi: 10.5281/zenodo.18931200.

[14]. R. M. Arienth, A. M, G. S. K. Reddy, and T. Sampath, "An IoT-Based Smart Irrigation System for Indian Agriculture: A Region-Specific Model with Sensor Integration and AI Scheduling," pp. 1079–1084, Jan. 2026, doi: 10.1109/icmcsi67283.2026.11412541.

[15]. V. H. Kalmani, N. V. Dharwadkar, and V. Thapa, "Crop Yield Prediction using Deep Learning Algorithm based on CNN-LSTM with Attention Layer and Skip Connection," Indian Journal of Agricultural Research, Dec. 2024, doi: 10.18805/ijare. a-6300.

[16]. S. Somvanshi et al., "From Tiny Machine Learning to Tiny Deep Learning: A Survey," Nov. 13, 2025, Cornell University. doi: 10.1145/3776588.

[17]. M. T. Lê, P. Wolinski, and J. Arbel, "Efficient Neural Networks for Tiny Machine Learning: A Comprehensive Review," HAL (Le Centre pour la Communication Scientifique Directe), vol. 17, no. 4, pp. 1–41, Feb. 2026, doi: 10.1145/3798276.

[18]. Y. Abadade, A. Temouden, H. Bamoumen, N. Benamar, Y. Chtouki, and A. Hafid, "A Comprehensive Survey on TinyML," IEEE Access, vol. 11, pp. 96892–96922, Jan. 2023, doi: 10.1109/access.2023.3294111.

[19]. V. Rajapakse, I. Karunanayake, and N. Ahmed, "Intelligence at the Extreme Edge: A Survey on Reformable TinyML," ACM Computing Surveys, vol. 55, pp. 1–30, Feb. 2023, doi: 10.1145/3583683.

[20]. D. Donskoy, V. Gvindjiliya, and E. Ivliev, "TinyML Classification for Agriculture Objects with ESP32," Digital, vol. 5, no. 4, pp. 48–48, Oct. 2025, doi: 10.3390/digital5040048.

[21]. A. R. Alba, J. F. Villaverde, A. Lacara, J. Domingo, and D. Aguirre, "Optimized FLOPs-Aware Knowledge Distillation for TinyML Applications in Agriculture," pp. 1–6, Feb. 2025, doi: 10.1109/icadeis65852.2025.10933276.

[22]. R. Gupta, V. Sanghvi, and S. Shabista, "TinyFusion-NAS: Hardware-Aware Neural Architecture Search with Lightweight Attention for Multimodal Sensor Fusion," pp. 1–2, Jan. 2026, doi: 10.1109/ccnc65079.2026.11366556.

[23]. A. K. Sinha, R. Mishra, and A. Gupta, "Adaptive Post-Training Quantization for Device-Aware TinyML Deployment," pp. 1–6, Nov. 2025, doi: 10.1109/cins67018.2025.11412109.

[24]. M. Ohamouddou, S. Ohamouddou, A. E. Afia, and R. Lasri, "ATMS-KD: Adaptive temperature and mixed sample knowledge distillation for a lightweight residual CNN in agricultural embedded systems," Smart Agricultural Technology, vol. 12, pp. 101617–101617, Nov. 2025, doi: 10.1016/j.atech.2025.101617.

[25]. X. Lin, S. Chen, N. A. Yan, J. Pi, T. Zhu, and L. Xu, "Knowledge distillation for smart agriculture: methods, applications, and future directions," Smart Agricultural Technology, vol. 14, pp. 102019–102019, Mar. 2026, doi: 10.1016/j.atech.2026.102019.

[26]. X. Zhang, Q. Feng, D. Zhu, X. Liang, and J. Zhang, "Compressing recognition network of cotton disease with spot-adaptive knowledge distillation," Frontiers in Plant Science, vol. 15, Sep. 2024, doi: 10.3389/fpls.2024.1433543.

[27]. Y. Hui, F. Xie, D. Long, Y. Long, P. Yu, and H. Chen, "An accurate irrigation volume prediction method based on an optimized LSTM model," PubMed Central, vol. 10, Jun. 2024, doi: 10.7717/peerj-cs.2112.

[28]. D. Kaplun et al., "An intelligent agriculture management system for rainfall prediction and fruit health monitoring," Scientific Reports, vol. 14, no. 1, pp. 512–512, Jan. 2024, doi: 10.1038/s41598-023-49186-y.

[29]. Y. Wang and Y. Zha, "Comparison of transformer, LSTM and coupled algorithms for soil moisture prediction in shallow-groundwater-level areas with interpretability analysis," Agricultural Water Management, vol. 305, pp. 109120–109120, Oct. 2024, doi: 10.1016/j.agwat.2024.109120.

[30]. Y. Song, L. Zhang, Y. Yan, Z. Li, M. Yang, and Z. Hu, "Time-Frequency Attention Fusion Network with Wavelet-Transformer for Time Series Classfication," pp. 141–145, Nov. 2025, doi: 10.1109/ic-nidc67200.2025.11390192.

[31]. Z. Xiao, J. Liu, X. Cao, K. Wang, D. Li, and Q. Liu, "Time-Series Forecasting Method Based on Hierarchical Spatio-Temporal Attention Mechanism," Sensors, vol. 25, no. 13, pp. 4001–4001, Jun. 2025, doi: 10.3390/s25134001.

[32]. A. Haboub, H. Baali, and A. Bouzerdoum, "AttDCT: Attention-Based Deep Learning

Approach for Time Series Classification in the DCT Domain," IEEE Transactions on Artificial Intelligence, vol. 6, no. 6, pp. 1626–1638, Jan. 2025, doi: 10.1109/tai.2025.3534141.

[33]. S. K. H. and K. T. Veeramanju, "Predictive Models for Optimal Irrigation Scheduling and Water Management: A Review of AI and ML Approaches," International Journal of Management Technology and Social Sciences, pp. 94–110, May 2024, doi: 10.47992/ijmts.2581.6012.0346.

[34]. S. Ahmad, F. Siddiqui, Dr. Y. Perwej, H. Rizvi, and Dr. N. Akhtar, "Modelling Crop Yield with Deep CNN-LSTM for Spatiotemporal Data Analysis," International Journal of Scientific Research in Computer Science Engineering and Information Technology, vol. 11, no. 5, pp. 54–64, Sep. 2025, doi: 10.32628/cseit25111697.

[35]. K. Jaswal, S. R. Bhambak, M. K. GUJAR, S. Mohite, S. ANANTHARAMAN, and S. BHAGYALAKSHMY, "Development of 1961-1990 monthly surface climatology of India and patterns of differences of some meteorological parameters with respect to the 1951-1980 climatology," MAUSAM, vol. 63, no. 3, pp. 377–390, Jul. 2012, doi: 10.54302/mausam. v63i3.1222.

[36]. J. Guiguitant et al., "Relevance of limited-transpiration trait for lentil (Lens culinaris Medik.) in South Asia," Field Crops Research, vol. 209, pp. 96–107, May 2017, doi: 10.1016/j.fcr.2017.04.013.

[37]. N. Devineni, S. Perveen, and U. Lall, "Solving groundwater depletion in India while achieving food security," Nature Communications, vol. 13, no. 1, pp. 3374–3374, Jun. 2022, doi: 10.1038/s41467-022-31122

[38]. R. G. Allen, L. S. Pereira, D. Raes, and M. Smith, Crop evapotranspiration: guidelines for computing crop water requirements, no. 1. 1998.

[39]. R. Chakraborti, K. F. Davis, R. DeFries, N. D. Rao, J. Joseph, and S. Ghosh, "Crop switching for water sustainability in India's food bowl yields co-benefits for food security and farmers' profits," Nature Water, vol. 1, no. 10, pp. 864–878, Oct. 2023, doi: 10.1038/s44221-023-00135-z.

[40]. C. Banbury et al., "MLPerf Tiny Benchmark," Jun. 14, 2021, Cornell University. doi: 10.48550/arxiv.2106.07597.