International Journal of Wireless and Ad Hoc Communication

♦ ISSN: 2692-4056 Vol: 10, No. 021, 2026 ♦ PP. 43–52

Article DOI: https://doi.org/10.54216/IJWAC.100107




FULL LENGTH ARTICLE

# Time-Forward Evaluation of Denial-of-Service Detection in LEACH-Based Wireless Sensor Networks: Drift, Feature Budgets and Decision Costs

**Firas Zobary**[1,*]

[1] School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan, Hubei, China

**Email:** firaszobary@whut.edu.cn

Received: February 04, 2026 Revised: March 11, 2026 Accepted: April 22, 2026 ⋆ Corresponding author

## ABSTRACT

Intrusion detectors for cluster-based wireless sensor networks are scored on shuffled records, although a deployed detector is trained on the early rounds of a network and consulted for the rest of its life. This paper re-examines denial-of-service detection on the WSN-DS benchmark under four protocols, the strictest of which trains on the first part of the simulation and tests on what follows. Five classifiers are compared, class-conditional drift is measured with the Kolmogorov–Smirnov distance, and a feature budget is chosen by forward selection on a validation window. Random and block-grouped folds agree to within half a point of macro-F1. The time-forward split does not: with all fifteen behavioural features a random forest falls from 0.974 to between 0.833 and 0.946 depending on its seed, because most late Blackhole and Grayhole records lie outside the training range. A six-feature forest restores 0.959 with negligible seed variance and cuts the report payload by 60%. Isotonic calibration followed by expectation-maximisation prior correction tracks the rising attack share to within one point and lowers the balanced decision cost by 38%, yet raises it at high miss penalties. Validation scores near the training period could not tell the fragile configuration from the stable one.

**Keywords:** Wireless sensor networks ▪ LEACH ▪ Denial-of-service detection ▪ Dataset shift ▪ Probability calibration

## 1. INTRODUCTION

Wireless sensor networks are built from nodes that have little energy and less memory, and the protocols that organise them were designed to spend both carefully [1]. LEACH is the best-known example. It divides the lifetime of the network into rounds, elects a new set of cluster heads (CHs) in each round by a randomised threshold, lets every other node join the nearest head, and has the head hand out a time-division schedule, collect the data of its members and forward an aggregate to the base station (BS) [2]. The rotation spreads the cost of long-range transmission. It also hands a great deal of trust to whichever node announces itself as a head, and a compromised node that does so can drop the traffic of a whole cluster, drain its neighbours with repeated advertisements or issue a schedule that makes them collide [3, 4, 5].

Learning-based intrusion detection is the usual answer [6], and for LEACH the WSN-DS dataset of Almomani et al. [7] has become the common benchmark: per-node counters of the set-up and steady-state phases, simulated in NS-2, with four attacks (Blackhole, Grayhole, Flooding and the scheduling attack that the file labels TDMA). Reported scores are high. The original neural network reached class-wise accuracies between 75.6% and 99.8% under ten-fold cross-validation [7], and later studies with random forests, convolutional networks and resampling pipelines report overall accuracy near or above 99% [8, 9, 10]. All of them shuffle the records

© 2026 The Author(s). Published by ASPG. This article is licensed under CC BY 4.0.

before splitting.

A deployed detector does not get shuffled data. It is fitted on what the network produced up to some time $t_0$ and then applied to every later round, by which point nodes have died, clusters have shrunk and the attacker's footprint in the counters has changed. Let $P$ denote the joint distribution of feature vector $\mathbf{x}$ and class $y$ before $t_0$, $Q$ the distribution after it, and $R_D(h) = \mathbb{E}_{(\mathbf{x},y)\sim D}\,\mathbf{1}[h(\mathbf{x}) \neq y]$ the risk of a classifier $h$ under $D$. Shuffled evaluation estimates $R_{P\cup Q}(h)$ for an $h$ that has seen both periods, whereas operation depends on

$$\Delta(h) = R_Q(h_P) - R_P(h_P), \qquad h_P \text{ fitted on } P \text{ only}, \tag{1}$$

and nothing in a shuffled score bounds $\Delta$. How large it is on WSN-DS has, to our knowledge, not been measured.

This paper measures it. We compare four evaluation protocols on the same records, trace the loss under the time-forward protocol to drift in the class-conditional distributions of two attack classes, and show that the loss depends on the random seed of the ensemble to a degree that validation data cannot expose. Two remedies are then tested on held-out time: a feature budget, which is also what a sensor node would want for its report payload, and probability calibration with a correction for the changing attack prior, which is what a cost-sensitive alarm threshold needs. Section 2 reviews related work, Section 3 sets out the data, protocols and estimators, Section 4 reports the results with their limits, and Section 5 concludes.

## 2. RELATED WORK

### 2.1 Attacks on LEACH and detectors trained on WSN-DS

Karlof and Wagner [3] catalogued routing-layer attacks on sensor networks before most of the protocols had been implemented, and the denial-of-service taxonomy of Wood and Stankovic [4], later extended by Raymond and Midkiff [5], still covers the four behaviours that WSN-DS contains. Butun et al. [6] survey the detection side and point out that the detector must respect the same energy limits as the network it protects. WSN-DS made supervised comparison possible [7]. Alsulaiman and Al-Ahmadi [8] evaluated several classical learners on it and ranked the random forest first; Salmi and Oughdir [9] trained dense, convolutional and recurrent networks on a random 80/20 split and report 98.79% accuracy for the convolutional model; Talukder et al. [10] combined SMOTE-Tomek resampling with tree ensembles. Scores of this kind leave little room for improvement. They say less about transfer, because each one comes from a partition in which training and test records are drawn from the same simulated minutes.

### 2.2 Evaluation under shift

The wider intrusion-detection literature has been warning about this for some time. Sommer and Paxson [11] argued that network traffic violates the closed-world assumption behind most classifiers, and the survey of Buczak and Guven [12] lists the dependence on a few ageing datasets as a standing weakness. Arp et al. [13] found sampling bias and temporal snooping in many of the top-venue security papers they reviewed, while Pendlebury et al. [14] showed for malware that a detector with a high F1 under random folds can lose much of it when the test period follows the training period. Kapoor and Narayanan [15] document the same leakage patterns across seventeen scientific fields, and Roberts et al. [16] recommend blocking by the structure of the data whenever observations are grouped in time or space.

Shift itself comes in kinds [17, 18]. Under prior (or label) shift the class proportions move while the class-conditional densities stay put,

$$Q(\mathbf{x},y) = Q(y)\,P(\mathbf{x} \mid y), \tag{2}$$

and the posterior of a calibrated classifier can be corrected in closed form once $Q(y)$ is known or estimated [19, 20]. Where $P(\mathbf{x} \mid y)$ changes as well, no such correction exists, and D'Amour et al. [21] show that models which are indistinguishable on held-out data from the training distribution may then behave very differently from one another. Both situations turn up in the experiments below.

## 3. MATERIALS AND METHODS

### 3.1 Dataset and feature vector

WSN-DS [7] is distributed as a single CSV file, and the record and class counts of the copy used here match those its authors published. It holds one record per node per reporting instant of a 100-node LEACH network simulated for 3,600 s. Each record is

$$\begin{aligned} &\mathbf{r}_i = \big(\mathrm{id}_i,\ t_i,\ \mathrm{ch}_i,\ \mathbf{x}_i,\ y_i\big),\\ &\mathbf{x}_i \in \mathbb{R}^{15},\\ &y_i \in \mathscr{Y},\\ &\mathscr{Y} = \{\mathrm{Normal}, \mathrm{Blackhole}, \mathrm{Grayhole},\\ &\qquad \mathrm{Flooding}, \mathrm{TDMA}\}. \end{aligned} \tag{3}$$

where $\mathrm{id}_i$ is a node identifier that concatenates a stage-and-round prefix with a node index from 0 to 99, $t_i$ is the simulation time and $\mathrm{ch}_i$ the identifier of the node's cluster head. The fifteen behavioural features in $\mathbf{x}_i$ are the CH flag (`Is_CH`), two distances (`Dist_To_CH`, `dist_CH_To_BS`), the counts of advertisement, join-request and schedule messages sent and received (`ADV_S`, `ADV_R`, `JOIN_S`, `JOIN_R`, `SCH_S`, `SCH_R`), the node's rank in the schedule, the data packets sent, received and forwarded to the BS (`DATA_S`, `DATA_R`, `Data_Sent_To_BS`), the cluster sending code and the energy consumed (spelled `Expaned_Energy` in the file). We keep the file's names. Integer division of $\mathrm{id}_i$ by 1,000 yields 161 distinct prefixes, which we call blocks; the time field takes 196 distinct values.

The file needed no imputation, but it is not clean. It contains 8,873 fully duplicated rows, and 41,209 rows repeat a behavioural vector that has already occurred; 69 behavioural vectors carry more than one label. Nothing was removed, since a detector in the field would meet the same repeats. Table 1 gives the class composition and Figure 1 shows how it moves: the volume of records falls as nodes die, Flooding disappears after 1,500 s, and the attack share of a 300 s window climbs from under 3% to 39%.

**Table 1.** Class composition of WSN-DS and of the three time segments used by the time-forward protocol

| Class | All | Fit | Valid. | Test |
|---|---|---|---|---|
| Normal | 340,066 | 179,047 | 36,898 | 124,121 |
| Blackhole | 10,049 | 1,249 | 337 | 8,463 |
| Grayhole | 14,596 | 1,331 | 461 | 12,804 |
| Flooding | 3,312 | 1,819 | 860 | 633 |
| TDMA | 6,638 | 2,179 | 507 | 3,952 |
| Total | 374,661 | 185,625 | 39,063 | 149,973 |
| Attack share | 9.23% | 3.54% | 5.54% | 17.24% |

Time ranges (s): fit $t \leq 753$; validation $753 < t \leq 953$; test $t > 953$.

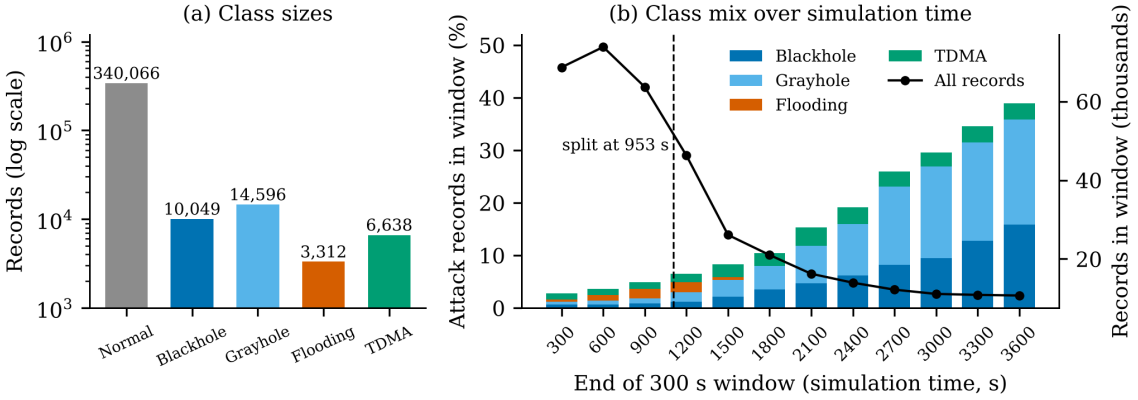



**Figure 1.** Class sizes (a) and class mix per 300 s window of simulation time (b). The dashed line marks the time-forward split.

### 3.2 Evaluation protocols

Let $\mathcal{I} = \{1, \ldots, n\}$ index the records. A protocol is a family of train/test pairs $(\mathcal{T}_k, \mathcal{S}_k)$ with $\mathcal{T}_k \cap \mathcal{S}_k = \emptyset$, together with the fields the classifier may read. Four are used.

P1 is the common practice: five stratified random folds, all 18 fields including id, $t$ and ch. P2 uses the same folds and only $\mathbf{x}$. P3 keeps $\mathbf{x}$ and draws five stratified folds over blocks, so that with $b(i)$ the block of record $i$,

$$\{b(i) : i \in \mathcal{T}_k\} \cap \{b(j) : j \in \mathcal{S}_k\} = \emptyset \quad \text{for every fold } k. \tag{4}$$

P4, the time-forward protocol, has a single pair,

$$\mathcal{T} = \{i : t_i \leq t_0\}, \qquad \mathcal{S} = \{i : t_i > t_0\}, \qquad t_0 = 953 \text{ s}, \tag{5}$$

which puts 60% of the records in training. Whenever a choice has to be made (feature subset, model family, calibration map), $\mathcal{T}$ is itself cut at 753 s into a fit segment and a validation segment (Table 1); the test segment is not consulted.

How much a protocol lets the classifier recognise records it has already seen is measured by the overlap rate

$$\omega = \frac{1}{|\mathcal{S}|} \sum_{j \in \mathcal{S}} \mathbf{1}\big[\exists i \in \mathcal{T} : \mathbf{x}_i = \mathbf{x}_j\big], \tag{6}$$

averaged over folds and also computed per class.

### 3.3 Classifiers and scores

Five classifiers from scikit-learn [22] are compared at fixed, untuned settings. Multinomial logistic regression (LR) on standardised features models

$$\Pr(y = c \mid \mathbf{x}) = \frac{\exp(\mathbf{w}_c^\top \mathbf{x} + b_c)}{\sum_{c' \in \mathcal{Y}} \exp(\mathbf{w}_{c'}^\top \mathbf{x} + b_{c'})}, \tag{7}$$

and Gaussian naive Bayes (NB) is included as a second low-cost baseline. The decision tree (DT, at least five records per leaf) splits a node $m$ with class proportions $p_{mc}$ on the feature and threshold that most reduce the Gini impurity

$$G(m) = 1 - \sum_{c \in \mathcal{Y}} p_{mc}^2. \tag{8}$$

The random forest (RF) averages $B = 100$ such trees grown on bootstrap samples with random feature subsets [23],

$$\hat{p}_c(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^{B} \hat{p}_c^{(b)}(\mathbf{x}), \qquad \hat{y} = \arg\max_c \hat{p}_c(\mathbf{x}), \tag{9}$$

and the histogram gradient-boosting classifier (HGB), a LightGBM-style learner [24, 25], builds an additive score per class over $M = 200$ stages with learning rate $\nu = 0.1$,

$$F_c^{(m)}(\mathbf{x}) = F_c^{(m-1)}(\mathbf{x}) + \nu\, g_c^{(m)}(\mathbf{x}), \tag{10}$$

each $g_c^{(m)}$ being a regression tree fitted to the negative gradient of the multinomial deviance on features binned into at most 255 levels. The three tree-based models are randomised. Under P4 each is therefore refitted with five seeds, and we report the mean, the standard deviation and the range.

With 91% of the records Normal, accuracy says little [26].

The headline score is the macro-averaged F1 [27],

$$\text{macro-F1} = \frac{1}{|\mathscr{Y}|}\sum_{c\in\mathscr{Y}}\frac{2\,\mathrm{TP}_c}{2\,\mathrm{TP}_c+\mathrm{FP}_c+\mathrm{FN}_c}, \tag{11}$$

supported by per-class recall and by the multi-class Matthews correlation coefficient [28], which for a confusion matrix $\mathbf{C}$ with row sums $a_c$, column sums $e_c$, trace $s$ and total $n$ is

$$\mathrm{MCC} = \frac{s\,n-\sum_c a_c e_c}{\sqrt{\left(n^2-\sum_c e_c^2\right)\left(n^2-\sum_c a_c^2\right)}}. \tag{12}$$

Intervals under P4 come from a block bootstrap [29]: the 142 blocks present in the test segment are resampled with replacement 500 times and the 2.5th and 97.5th percentiles of the recomputed score are taken.

### 3.4 Drift measurement

For feature $j$ and class $c$, let $F^{P}_{jc}$ and $F^{Q}_{jc}$ be the empirical distribution functions of $x_j$ among the class-$c$ records before and after $t_0$. Drift is summarised by the two-sample Kolmogorov–Smirnov distance [30]

$$D_{jc} = \sup_{u}\left|F^{P}_{jc}(u)-F^{Q}_{jc}(u)\right| \in [0,1], \tag{13}$$

used here as a descriptive distance. With samples of this size every non-zero value would be significant, so no test is reported. A second, cruder quantity is the share of test records of class $c$ with one or more coordinates outside the interval spanned by the training records of that class,

$$\rho_c = \frac{1}{|\mathscr{S}_c|}\sum_{i\in\mathscr{S}_c}\mathbf{1}\Big[\exists\, j:\; x_{ij}\notin\big[\min_{l\in\mathscr{T}_c}x_{lj},\;\max_{l\in\mathscr{T}_c}x_{lj}\big]\Big], \tag{14}$$

which matters for trees in particular, since a tree assigns every point beyond its outermost threshold to whatever leaf happens to lie there.

### 3.5 Feature budget and reporting energy

A greedy wrapper [31] orders the features. Starting from $\mathscr{F}_0=\emptyset$, step $k$ adds

$$j_k = \operatorname*{arg\,max}_{j\notin\mathscr{F}_{k-1}} V\big(\mathscr{F}_{k-1}\cup\{j\}\big), \qquad \mathscr{F}_k=\mathscr{F}_{k-1}\cup\{j_k\}, \tag{15}$$

where $V(\mathscr{F})$ is the validation macro-F1 of a decision tree fitted on the fit segment with features $\mathscr{F}$; a single tree serves as the scorer because it is deterministic and cheap, which matters in a loop of 120 fits. The budget is then the smallest subset whose score comes within a tolerance of the best one observed,

$$k^{*} = \min\Big\{k:\; V(\mathscr{F}_k)\ge\max_{k'}V(\mathscr{F}_{k'})-\varepsilon\Big\}, \qquad \varepsilon=0.001. \tag{16}$$

What a report costs the radio is priced with the first-order model that LEACH itself uses [2], under which sending $\ell$ bits over distance $d$ costs

$$E_{\mathrm{tx}}(\ell,d) = \begin{cases}\ell E_{\mathrm{elec}}+\ell\,\varepsilon_{\mathrm{fs}}\,d^2, & d<d_0,\\ \ell E_{\mathrm{elec}}+\ell\,\varepsilon_{\mathrm{mp}}\,d^4, & d\ge d_0,\end{cases} \qquad d_0=\sqrt{\varepsilon_{\mathrm{fs}}/\varepsilon_{\mathrm{mp}}}, \tag{17}$$

with $E_{\mathrm{elec}}=50$ nJ/bit, $\varepsilon_{\mathrm{fs}}=10$ pJ/bit/m$^2$ and $\varepsilon_{\mathrm{mp}}=0.0013$ pJ/bit/m$^4$, so that $d_0\approx 87.7$ m. For a report of $k$ features of $\beta$ bits each, the mean energy over the distances actually recorded in the dataset (`Dist_To_CH` for a member, `dist_CH_To_BS` for a head) is

$$\bar{E}(k) = \frac{k\beta}{n}\sum_{i=1}^{n}E_{\mathrm{tx}}(1,d_i). \tag{18}$$

We take $\beta=16$.

### 3.6 Calibration, prior correction and decision cost

Alarms are binary. The five classes are collapsed to attack ($a=1$) against Normal, with score $s(\mathbf{x})=1-\hat{p}_{\mathrm{Normal}}(\mathbf{x})$ from a model trained on the fit segment. Isotonic regression [32, 33] on the validation segment $\mathscr{V}$ gives the calibrated probability

$$q(\mathbf{x}) = \hat{m}\big(s(\mathbf{x})\big), \qquad \hat{m} = \operatorname*{arg\,min}_{m\ \text{non-decreasing}}\sum_{i\in\mathscr{V}}\big(a_i-m(s_i)\big)^2. \tag{19}$$

This probability is calibrated for the validation prior $\pi_0$ (5.54%) and for no other. If the test period obeys Eq. (2) with prior $\pi$, the posterior becomes [19]

$$\tilde{q}(\mathbf{x};\pi) = \frac{\dfrac{\pi}{\pi_0}\,q(\mathbf{x})}{\dfrac{\pi}{\pi_0}\,q(\mathbf{x})+\dfrac{1-\pi}{1-\pi_0}\,\big(1-q(\mathbf{x})\big)}, \tag{20}$$

and $\pi$ can be estimated without labels by iterating the expectation-maximisation update

$$\pi^{(r+1)} = \frac{1}{|\mathscr{W}|}\sum_{i\in\mathscr{W}}\tilde{q}\big(\mathbf{x}_i;\pi^{(r)}\big), \qquad \pi^{(0)}=\pi_0, \tag{21}$$

to a fixed point over a set $\mathscr{W}$ of unlabelled records. Two choices of $\mathscr{W}$ are tried: the whole test segment (global), and each 300 s window on its own (windowed), which is the version a base station could run online because it needs nothing beyond the records of the last five minutes.

An alarm is raised when the probability exceeds the Bayes threshold for a false-negative cost $c_{\mathrm{FN}}$ and a false-positive cost $c_{\mathrm{FP}}$ [34],

$$\hat{a}(\mathbf{x}) = \mathbf{1}\big[\tilde{q}(\mathbf{x})\ge\tau^{*}\big], \qquad \tau^{*} = \frac{c_{\mathrm{FP}}}{c_{\mathrm{FP}}+c_{\mathrm{FN}}} = \frac{1}{1+\kappa}, \qquad \kappa=\frac{c_{\mathrm{FN}}}{c_{\mathrm{FP}}}, \tag{22}$$

and the outcome is scored by the cost per record in units of $c_{\mathrm{FP}}$,

$$C(\kappa) = \frac{\kappa\,\mathrm{FN}+\mathrm{FP}}{|\mathscr{S}|}. \tag{23}$$

Probability quality is reported as the Brier score [35] and the

expected calibration error over $K = 15$ equal-width bins $B_k$ [36],

$$\begin{aligned} \mathrm{BS} &= \frac{1}{|\mathscr{S}|}\sum_{i\in\mathscr{S}}\left(\tilde{q}_i - a_i\right)^2, \\ \mathrm{ECE} &= \sum_{k=1}^{K}\frac{|B_k|}{|\mathscr{S}|} \\ &\quad \left|\frac{1}{|B_k|}\sum_{i\in B_k} a_i - \frac{1}{|B_k|}\sum_{i\in B_k}\tilde{q}_i\right|, \end{aligned} \tag{24}$$

together with the negative log-likelihood (NLL).

## 4. RESULTS AND DISCUSSION

### 4.1 Overlap and the three shuffled protocols

Table 2 reports the overlap rate of Eq. (6). Under random folds 12.8% of the held-out records have an exact twin in training, and for Blackhole the figure is 62.9%: the attacker's counters take few distinct values in the early network. Grouping by block lowers the rate only to 10.0%. The time-forward split brings it down to 6.9% overall and to 6.9% and 1.8% for Blackhole and Grayhole, the first sign that the late attack records are not the early ones seen again.

The three shuffled protocols barely differ in score (Table 3, Figure 2). Dropping the identifier fields costs the random forest 0.007 of macro-F1 (0.980 under P1, 0.974 under P2), and blocking costs a further 0.004. We had expected more from P3. That so little is lost shows that the published near-ceiling figures are not produced by identifier leakage or by duplicated rows in any large measure; a tree ensemble learns the early and middle network well from the fifteen counters alone. Naive Bayes is the one model that the identifiers hurt, its macro-F1 rising from 0.551 to 0.774 once they are removed.

### 4.2 The time-forward protocol and seed dependence

P4 is another matter. Accuracy still looks respectable, between 0.951 and 0.978 for the tree-based models, because 83% of the test records are Normal and Normal recall stays at 0.999, but macro-F1 does not: the random forest drops from 0.974 to a seed mean of 0.881 and the boosted model from 0.962 to 0.844, so that the gap of Eq. (1), read on macro-F1, is 0.093 for the model that led every shuffled comparison and 0.118 for the boosted one. The ranking inverts. A single decision tree, second of the three under P2, is first under P4 at 0.935.

The means hide the more troubling result, which Figure 3 displays. Nothing but the seed separates the five random forests, and their macro-F1 values are 0.833, 0.835, 0.847, 0.946 and 0.943; the five boosted models give 0.941, 0.829, 0.842, 0.784 and 0.828. The first experiment we ran used one seed and happened to draw the best boosted model and the worst forest, which would have supported a confident and wrong conclusion about the two families. The decision tree does not move (standard deviation 0.0002). Reordering the columns of the feature matrix was enough to shift one boosted model from 0.941 to 0.943, a small change, but one that a deterministic procedure would not show.

Where the loss occurs is plain from the confusion matrices in Figure 4. Flooding and TDMA recall slip by a few points. The real damage is in the Blackhole row: in the worst forest 5,537 of the 8,463 late Blackhole records are labelled Grayhole (recall 0.343), and across seeds Blackhole recall ranges from 0.343 to 0.873 for the forest and from 0.310 to 0.958 for the boosted model (Table 4). The alarm itself is rarely missed, since a Blackhole called Grayhole is still an attack, yet the two call for different responses, and a detector whose attribution depends on its seed is hard to act on.

### 4.3 Class-conditional drift

Prior shift alone cannot be responsible, because a change in class proportions leaves a classifier's ranking of two attack classes untouched. Figure 5 shows what else changed. For Blackhole and Grayhole the Kolmogorov–Smirnov distance of Eq. (13) between the two periods lies between 0.64 and 0.77 on five counters (`ADV_R`, `JOIN_R`, `SCH_S`, `DATA_R` and `Expaned_Energy`), while for Flooding no feature exceeds 0.25. The direction is what an ageing LEACH network would produce. A Blackhole head received 20.2 join requests on average before 953 s and 2.4 after it; the data packets it received fell from 951 to 185, and the advertisements it heard from other heads rose from 4.8 to 19.4, as fewer live members were shared among more self-declared heads. Grayhole moved the same way (join requests 17.0 to 1.1, packets received 853 to 85).

The range statistic of Eq. (14) makes the consequence concrete: 66.5% of the late Blackhole records and 73.9% of the late Grayhole records have one or more features outside anything their class showed in training, against 2.0% for TDMA and none for Flooding. In those regions the training data do not determine the boundary between the two classes, so each tree settles it by whichever threshold its bootstrap sample and feature draw produced and an ensemble inherits the accident from its seed. This is the underspecification that D'Amour et al. [21] describe. Which counter carries the instability the experiment only partly identifies; Section 4.D narrows it down.

### 4.4 Feature budget

Forward selection by Eq. (15) entered the features in the order shown in Figure 6(a). Validation macro-F1 rose from 0.394 with `Expaned_Energy` alone to 0.921 with six features and then stayed within 0.001 of its maximum of 0.922, so Eq. (16) sets $k^* = 6$: `Expaned_Energy`, `SCH_S`, `ADV_S`, `Data_Sent_To_BS`, `JOIN_R` and `DATA_R`. On validation the full and the reduced forest are indistinguishable, at $0.928 \pm 0.002$ and $0.927 \pm 0.000$.

On the test segment they are not (Table 4). The six-feature forest, RF-6, scores 0.959 in every one of its five fits (range 0.9593 to 0.9598; block-bootstrap interval for seed 0 from 0.952 to 0.966), with Blackhole recall at 0.945 and Grayhole recall at 0.912. Under random and block-grouped folds it gives up almost nothing (0.972 and 0.968 against 0.974 and 0.970 for the full forest); the boosted model improves as well, to 0.940, but keeps a seed spread of 0.015. The budget is no drift filter. Two of the six retained features, `JOIN_R` and `DATA_R`, are among those that drift most. What it discards is `ADV_R`: a boosted model on the first nine selected features scored 0.925 on one seed and, once `ADV_R` was added as the tenth, 0.736. One seed is thin evidence. A leave-one-feature-

**Table 2.** Share of held-out records whose behavioural vector also occurs in the training set (overlap rate $\omega$, %)

| Protocol | All | Normal | Blackhole | Grayhole | Flooding | TDMA |
|---|---|---|---|---|---|---|
| Random folds (P1, P2) | 12.8 | 10.3 | 62.9 | 32.5 | 8.0 | 20.1 |
| Block-grouped folds (P3) | 10.0 | 8.0 | 58.1 | 22.3 | 0.1 | 17.7 |
| Time-forward (P4) | 6.9 | 7.6 | 6.9 | 1.8 | 0.3 | 3.5 |

**Table 3.** Scores by protocol. P1 to P3: mean over five folds. P4: single split; for DT, RF and HGB the mean over five seeds, with the standard deviation of macro-F1

| | P1 | P2 (random, 15 features) | | | P3 (block-grouped) | | | P4 (time-forward) | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Model | F1 | Acc. | F1 | MCC | Acc. | F1 | MCC | Acc. | F1 | MCC |
| LR | 0.849 | 0.974 | 0.851 | 0.854 | 0.974 | 0.850 | 0.853 | 0.947 | 0.838 | 0.827 |
| NB | 0.551 | 0.954 | 0.774 | 0.770 | 0.954 | 0.774 | 0.770 | 0.634 | 0.356 | 0.346 |
| DT | 0.974 | 0.995 | 0.967 | 0.972 | 0.994 | 0.962 | 0.966 | 0.978 | $0.935 \pm 0.000$ | 0.924 |
| RF | 0.980 | 0.996 | 0.974 | 0.977 | 0.995 | 0.970 | 0.973 | 0.962 | $0.881 \pm 0.053$ | 0.874 |
| HGB | 0.972 | 0.995 | 0.962 | 0.969 | 0.995 | 0.963 | 0.968 | 0.951 | $0.844 \pm 0.052$ | 0.831 |

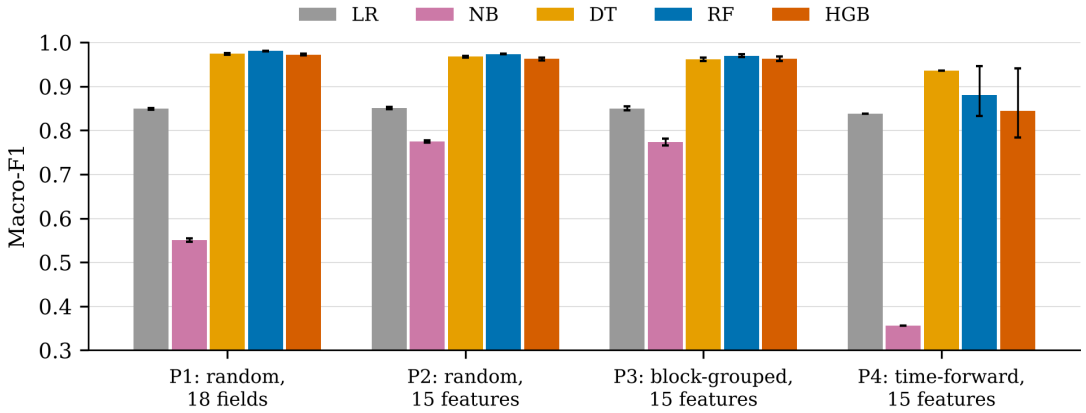



**Figure 2.** Macro-F1 by protocol and model. Error bars: fold standard deviation (P1 to P3); seed range (P4, tree-based models).

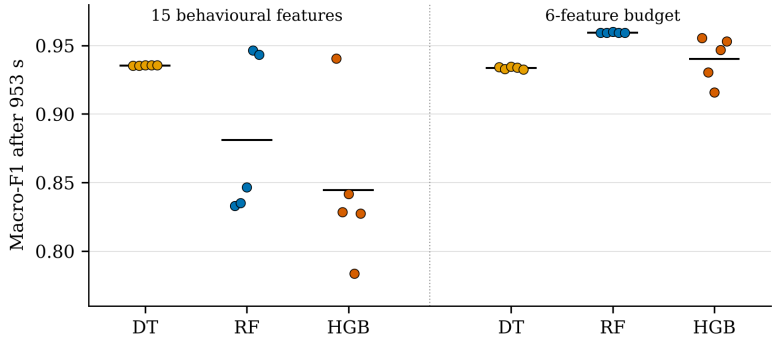



**Figure 3.** Time-forward macro-F1 of five independently seeded fits per model, with all 15 behavioural features and with the 6-feature budget. Bars mark the seed mean.

out run over all fifteen counters and several seeds would settle whether `ADV_R` is the sole carrier, and we have not done it.

The validation column of Table 4 deserves a second look, because it is all a practitioner would have had: it ranks RF-15 first by a margin smaller than its own seed spread and gives no warning of the 0.053 standard deviation that follows. A validation window 200 s wide, adjacent to the fit segment, shares its distribution too closely to expose the problem; the seed spread of the predictions themselves on unlabelled later data would be a cheaper alarm, and it is a diagnostic we would add to any deployment.

As for the node, Eq. (18) prices the saving. Members sit 25.6 m from their head on average and pay 63.9 nJ per bit; heads pay 282.9 nJ per bit toward the BS, 45.7% of their records lying beyond $d_0$. Weighted by the 88.4% share of member records this is 89.2 nJ per bit, so a 15-feature report of 240 bits costs 21.4 $\mu$J and the 6-feature report of 96 bits costs 8.6 $\mu$J. The payload shrinks by 60%. In absolute terms the energy is small. The argument for the budget rests on stability and on slot occupancy, and only marginally on joules. Model size

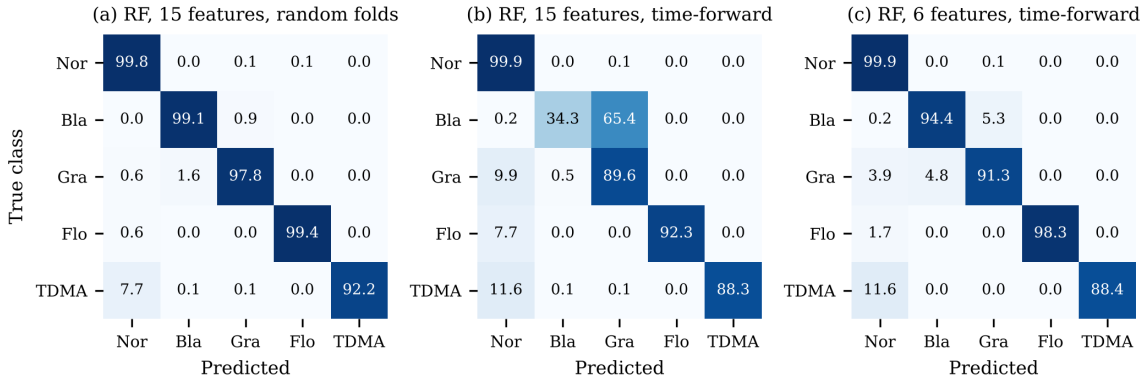



**Figure 4.** Row-normalised confusion matrices (%) of the random forest: pooled random folds (a); time-forward, 15 features, seed 0 (b); time-forward, 6 features, seed 0 (c).

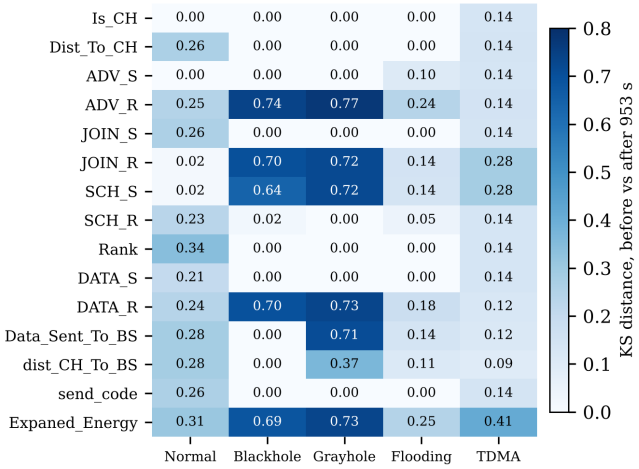



**Figure 5.** Kolmogorov–Smirnov distance between the distributions of each feature before and after 953 s, within each class. Zero entries are features that are constant for that class.

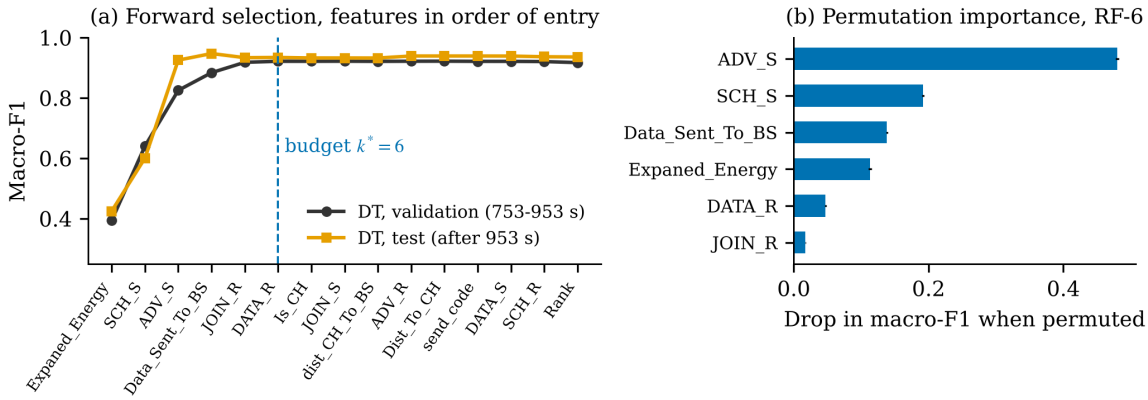



**Figure 6.** Forward selection on the validation window, with the test score of the same trees (a); permutation importance of the six selected features for RF-6 on the test segment (b).

points the same way for placement: RF-6 holds 184,916 tree nodes (18.8 MB serialised) and belongs at the base station, whereas DT-6, with 2,227 nodes and a macro-F1 of 0.934, could sit on a cluster head. Permutation importance for RF-6 (Figure 6(b)) is led by `ADV_S` (0.480), followed by `SCH_S` (0.192) and `Data_Sent_To_BS` (0.138).

### 4.5 Calibration, prior tracking and decision cost

Table 5 and Figure 7 report the binary alarm built on RF-6 fitted up to 753 s. Raw vote fractions are already sharp (Brier 0.0059). Isotonic calibration on the validation window made them worse on every measure, Brier rising to 0.0070, and Figure 7(a) shows why: the map was learned at a 5.5% attack prior and is applied where the prior is 17.2%, so mid-range scores are too low. The correction of Eq. (20) repairs this.

**Table 4.** Time-forward protocol: per-class recall and macro-F1 over five seeds, with the validation macro-F1 that was available when the choice was made. Brackets give the seed range of Blackhole recall

| | Recall on the test segment | | | | | Macro-F1 | |
|---|---|---|---|---|---|---|---|
| Model | Normal | Blackhole | Grayhole | Flooding | TDMA | Validation | Test |
| DT-15 | 0.999 | 0.944 [0.944, 0.944] | 0.824 | 0.830 | 0.890 | 0.917 ± 0.000 | 0.935 ± 0.000 |
| RF-15 | 0.999 | 0.566 [0.343, 0.873] | 0.893 | 0.935 | 0.883 | 0.928 ± 0.002 | 0.881 ± 0.053 |
| HGB-15 | 0.998 | 0.480 [0.310, 0.958] | 0.826 | 0.865 | 0.871 | 0.912 ± 0.004 | 0.844 ± 0.052 |
| DT-6 | 0.999 | 0.936 [0.936, 0.936] | 0.798 | 0.892 | 0.883 | 0.921 ± 0.000 | 0.934 ± 0.001 |
| RF-6 | 0.999 | 0.945 [0.944, 0.947] | 0.912 | 0.982 | 0.883 | 0.927 ± 0.000 | 0.959 ± 0.000 |
| HGB-6 | 0.999 | 0.945 [0.929, 0.975] | 0.859 | 0.965 | 0.841 | 0.904 ± 0.006 | 0.940 ± 0.015 |

**Table 5.** Binary alarm on the test segment (RF-6, mean over five seeds). $C(\kappa)$ is cost per 100 records at the threshold of Eq. (22).

| Method | Calibration metrics | | | Decision cost by $\kappa$ | | | | |
|---|---|---|---|---|---|---|---|---|
| | Brier | ECE | NLL | $C(1)$ | $C(5)$ | $C(10)$ | $C(25)$ | $C(50)$ |
| Raw vote fraction | 0.0059 | 0.0107 | 0.0262 | 0.75 | 1.44 | 2.22 | 4.33 | 7.51 |
| Isotonic | 0.0070 | 0.0116 | 0.0311 | 0.80 | 1.61 | 2.85 | 6.45 | 11.37 |
| Isotonic + global prior | 0.0045 | 0.0074 | 0.0243 | 0.49 | 1.51 | 2.74 | 5.43 | 8.36 |
| Isotonic + windowed prior | 0.0042 | 0.0080 | 0.0234 | 0.46 | 1.47 | 2.55 | 6.24 | 13.10 |

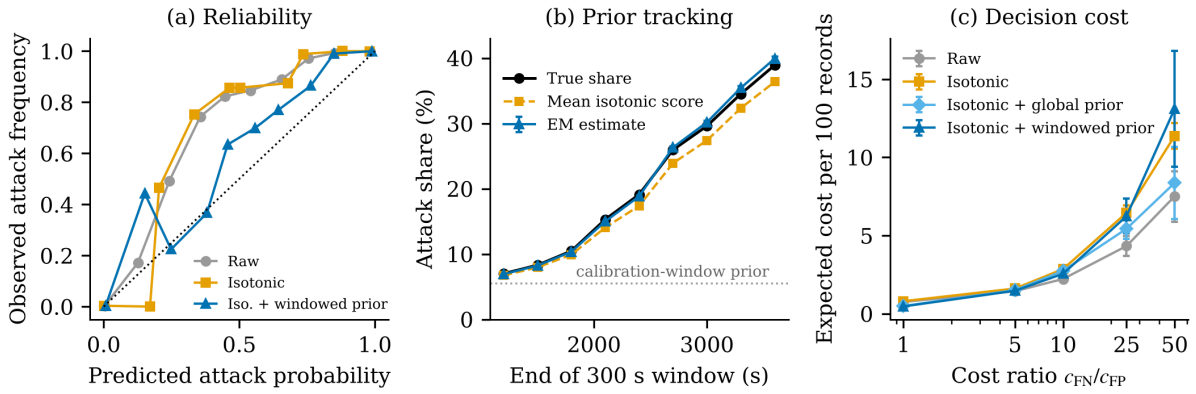



**Figure 7.** Reliability of the attack probability, seed 0 (a); true and EM-estimated attack share per 300 s window (b); decision cost against the cost ratio $\kappa$, with seed standard deviation (c).

With the global EM estimate the Brier score falls to 0.0045, and with windowed estimates to 0.0042, a 28% reduction against the raw score.

The prior estimates are the cleanest result of the study. Without a single test label, Eq. (21) returned a global attack share between 0.172 and 0.174 over five seeds against a true 0.172, and the windowed estimates follow the true share from 7.0% to 38.9% with a largest error of 1.0 percentage point, in the final window (Figure 7(b)). That the estimate works although Eq. (2) is violated for two classes is less surprising than it sounds: the binary alarm merges Blackhole and Grayhole, and their mutual confusion is invisible to it.

Better probabilities did not always buy better decisions. At $\kappa = 1$ the windowed correction cuts the cost per 100 records from 0.75 to 0.46, a 38% reduction, by halving the misses (991 to 505 on average) for 53 extra false alarms. At $\kappa = 5$ the four variants are within 0.17 of each other. Beyond that the raw score wins: at $\kappa = 50$ it costs 7.51 against 13.10 for the windowed correction, which raises 14,131 false alarms where the raw score raises 6,846. The threshold there is $\tau^* = 0.0196$, and a step function fitted on 2,165 validation attacks has few distinct levels that low, so the prior correction lifts a whole plateau of Normal records over the line at once. A smoother calibrator, or a validation window with more attacks, might remove the effect. We did not test either, and until someone does the practical advice is narrow: correct the prior when misses and false alarms cost about the same, and leave the raw score alone when misses are priced an order of magnitude higher.

### 4.6 Limitations

WSN-DS is one simulated network. Every statement here is about it, and a second topology, another node density or a physical testbed could show more drift or less. The split point of 953 s was fixed in advance to give a 60/40 partition and was never varied, although an earlier $t_0$ would leave fewer attack records to learn from and would probably widen the gap. No classifier was tuned. That is deliberate for a protocol comparison, but it means that nothing is claimed about the best attainable score. Five seeds are enough to show that the spread exists and too few to estimate its distribution. The block structure, lastly, was inferred from the identifier format; the dataset documentation does not say whether blocks correspond to independent simulation runs, and if they do not, P3 is a weaker test than its name suggests.

## 5. CONCLUSION

On WSN-DS, shuffled evaluation overstates what a detector trained on the early network will deliver later. The estimated gap for the two ensembles is

$$\begin{aligned} \hat{\Delta}_{\text{RF-15}} &= 0.974 - 0.881 = 0.093, \\ \hat{\Delta}_{\text{HGB-15}} &= 0.962 - 0.844 = 0.118 \end{aligned} \tag{25}$$

in macro-F1, and for both the seed alone moves the time-forward score by more than 0.11. Drift is the cause. As the network thins, the Blackhole and Grayhole counters change so far that two thirds or more of the late records of those classes fall outside the training range, whereas identifier leakage and duplicated rows, the usual suspects, account for about 0.01.

A six-feature forest chosen on a validation window removes the seed dependence, reaches 0.959 on held-out time and shortens the report by 60%. Unlabelled prior estimation by expectation-maximisation follows the attack share to within one percentage point and lowers the balanced decision cost by 38%; at high miss penalties the uncorrected score remains the better choice. The lesson for evaluation is specific: results on this benchmark should be reported on a time-forward split, over several seeds and with per-class recall, since none of the failures described here is visible in shuffled accuracy or in a validation window adjacent to the training data. Two questions are left. Whether the same budget holds on other LEACH variants we cannot say, nor which single counter is responsible for the instability.

## REFERENCES


[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” Computer Networks, vol. 38, no. 4, pp. 393-422, 2002.

[2] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” IEEE Transactions on Wireless Communications, vol. 1, no. 4, pp. 660-670, 2002.

[3] C. Karlof and D. Wagner, “Secure routing in wireless sensor networks: attacks and countermeasures,” Ad Hoc Networks, vol. 1, no. 2-3, pp. 293-315, 2003.

[4] A. D. Wood and J. A. Stankovic, “Denial of service in sensor networks,” Computer, vol. 35, no. 10, pp. 54-62, 2002.

[5] D. R. Raymond and S. F. Midkiff, “Denial-of-service in wireless sensor networks: attacks and defenses,” IEEE Pervasive Computing, vol. 7, no. 1, pp. 74-81, 2008.

[6] I. Butun, S. D. Morgera, and R. Sankar, “A survey of intrusion detection systems in wireless sensor networks,” IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 266-282, 2014.

[7] I. Almomani, B. Al-Kasasbeh, and M. Al-Akhras, “WSN-DS: a dataset for intrusion detection systems in wireless sensor networks,” Journal of Sensors, vol. 2016, Art. no. 4731953, 2016.

[8] L. Alsulaiman and S. Al-Ahmadi, “Performance evaluation of machine learning techniques for DOS detection in wireless sensor network,” International Journal of Network Security & Its Applications, vol. 13, no. 2, 2021, doi: 10.5121/ijnsa.2021.13202.

[9] S. Salmi and L. Oughdir, “Performance evaluation of deep learning techniques for DoS attacks detection in wireless sensor network,” Journal of Big Data, vol. 10, Art. no. 17, 2023.

[10] M. A. Talukder, S. Sharmin, M. A. Uddin, M. M. Islam, and S. Aryal, “MLSTL-WSN: machine learning-based intrusion detection using SMOTETomek in WSNs,” International Journal of Information Security, vol. 23, pp. 2139-2158, 2024.

[11] R. Sommer and V. Paxson, “Outside the closed world: on using machine learning for network intrusion detection,” in Proc. IEEE Symposium on Security and Privacy, pp. 305-316, 2010.

[12] A. L. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1153-1176, 2016.

[13] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, and K. Rieck, “Dos and don’ts of machine learning in computer security,” in Proc. 31st USENIX Security Symposium, pp. 3971-3988, 2022.

[14] F. Pendlebury, F. Pierazzi, R. Jordaney, J. Kinder, and L. Cavallaro, “TESSERACT: eliminating experimental bias in malware classification across space and time,” in Proc. 28th USENIX Security Symposium, pp. 729-746, 2019.

[15] S. Kapoor and A. Narayanan, “Leakage and the reproducibility crisis in machine-learning-based science,” Patterns, vol. 4, no. 9, Art. no. 100804, 2023.

[16] D. R. Roberts et al., “Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure,” Ecography, vol. 40, no. 8, pp. 913-929, 2017.

[17] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera, “A unifying view on dataset shift in classification,” Pattern Recognition, vol. 45, no. 1, pp. 521-530, 2012.

[18] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, “A survey on concept drift adaptation,” ACM Computing Surveys, vol. 46, no. 4, Art. no. 44, 2014.

[19] M. Saerens, P. Latinne, and C. Decaestecker, “Adjusting

the outputs of a classifier to new a priori probabilities: a simple procedure," Neural Computation, vol. 14, no. 1, pp. 21-41, 2002.

[20] Z. C. Lipton, Y.-X. Wang, and A. J. Smola, "Detecting and correcting for label shift with black box predictors," in Proc. 35th International Conference on Machine Learning, PMLR vol. 80, pp. 3122-3130, 2018.

[21] A. D'Amour et al., "Underspecification presents challenges for credibility in modern machine learning," Journal of Machine Learning Research, vol. 23, no. 226, pp. 1-61, 2022.

[22] F. Pedregosa et al., "Scikit-learn: machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[23] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.

[24] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," The Annals of Statistics, vol. 29, no. 5, pp. 1189-1232, 2001.

[25] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: a highly efficient gradient boosting decision tree," in Advances in Neural Information Processing Systems, vol. 30, pp. 3146-3154, 2017.

[26] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263-1284, 2009.

[27] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," Information Processing & Management, vol. 45, no. 4, pp. 427-437, 2009.

[28] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," BMC Genomics, vol. 21, Art. no. 6, 2020.

[29] B. Efron and R. J. Tibshirani, An Introduction to the Bootstrap. New York, NY, USA: Chapman & Hall, 1993.

[30] F. J. Massey Jr., "The Kolmogorov-Smirnov test for goodness of fit," Journal of the American Statistical Association, vol. 46, no. 253, pp. 68-78, 1951.

[31] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," Journal of Machine Learning Research, vol. 3, pp. 1157-1182, 2003.

[32] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in Proc. 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 694-699, 2002.

[33] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in Proc. 22nd International Conference on Machine Learning, pp. 625-632, 2005.

[34] C. Elkan, "The foundations of cost-sensitive learning," in Proc. 17th International Joint Conference on Artificial Intelligence, pp. 973-978, 2001.

[35] G. W. Brier, "Verification of forecasts expressed in terms of probability," Monthly Weather Review, vol. 78, no. 1, pp. 1-3, 1950.

[36] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in Proc. 34th International Conference on Machine Learning, PMLR vol. 70, pp. 1321-1330, 2017.