Sensor-Driven Training Feedback for Horses: A Machine Learning Engineering Approach

Sayidali Ibrahim Jama
Master of Applied IT
Fontys University of Applied Sciences
Eindhoven, The Netherlands
ibrahimsayidali10@gmail.com

***Abstract*—Professional equine trainers spend a significant amount of time manually writing feedback after every training session. This feedback is valuable, but it does not scale when a single expert is responsible for dozens of horses across a full competition yard. In this study we designed and implemented a system that reads raw sensor data and automatically produces structured training feedback for each session, without needing a human expert to write it every time.**

The system covers 62 horses and 2,476 recorded training sessions from a professional yard in the Netherlands. It combines a two-phase weak labeling strategy to handle the limited amount of hand-labeled data, a 99-feature engineering pipeline, three XGBoost classifiers trained under Leave-One-Horse-Out cross-validation, and SHAP-based explanations that make every prediction inspectable. The main status classifier reached an accuracy of 82.2 % and a macro F1 score of 0.800 under this strictest possible evaluation. The dominant predictive feature is the distance-based Acute:Chronic Workload Ratio (ACWR), followed by a composite session quality metric derived from heart rate training impulse and recovery score, whose combined SHAP weight exceeds the ACWR alone. The system correctly identifies the sessions most worth expert attention and produces traceable explanations that mirror the structure of expert reasoning.

The contribution is both a working system and a methodological reflection: we propose and evaluate a structured engineering approach to sensor-driven equine decision support, and distil the lessons into concrete recommendations.

***Index Terms*—equine training, ACWR, iTRIMP, XGBoost, SHAP, weak supervision, decision support, Leave-One-Horse-Out cross-validation**

I. Introduction

Equine training generates a large volume of sensor data after every session: GPS-derived distances broken down by gait, heart rate history and recovery, an integrated training impulse score (iTRIMP), rider assessments of how the horse felt and how stressed it seemed, and a workload ratio comparing this week's load against the past four weeks. In principle, all of that information should be enough to answer the question that an expert trainer answers manually after every session: is this horse on track, should we watch it closely, or does it need rest?

In practice, turning raw sensor data into a structured recommendation still requires a trained specialist. This study was conducted in collaboration with Carolien Munsters, an equine sports medicine specialist and researcher based in the Netherlands [1]. She is referred to hereafter as *the domain expert*, so that the findings and discussion generalise beyond this specific collaboration. For several years the domain expert has manually reviewed session data and written Dutch-language training notes in a commercial equine management platform. These notes cover load concerns, heart rate responses, recommended adjustments, and injury or fatigue flags. The problem is that this takes a great deal of time, the yard has 62 horses, and producing thorough notes for every session every day is not sustainable at scale.

In this study we investigated how a machine learning pipeline, built from a software engineering perspective, can automate this feedback reliably and transparently. We designed and implemented a complete prototype that ingests sensor data, engineers 99 numeric features, trains three XGBoost classifiers on weakly labeled data, and outputs a structured JSON decision block for each session. Every prediction carries a confidence score and a SHAP explanation [2] showing which sensor signals drove it. The goal is not to replace the domain expert's judgment entirely, but to produce a correct and inspectable first draft for every session so that expert attention can focus on the borderline cases where it adds the most value [3].

The main research question is:

> *How effectively can structured, traceable training feedback be automatically generated from equine sensor data alone?*

This breaks down into three sub-questions. **SQ1 (Feature relevance):** Which sensor-derived features are most predictive for each decision block? **SQ2 (Prediction accuracy):** How accurately do the predictions match expert-derived ground truth across different horses and training scenarios? **SQ3 (Interpretability):** Do the SHAP-based explanations align with how the domain expert describes her own reasoning, and do they provide traceable justifications for each prediction?

The remainder of the paper is organised as follows. Section II presents the dataset, the engineering approach, and the experimental design. Section III defines the evaluation metrics. Section IV reports the results. Section V discusses them and reflects on the engineering approach. Section VI concludes.

A. Related Work

The ACWR in sport and equine contexts. The Acute:Chronic Workload Ratio was introduced in human sport science [4] as a method for tracking whether an athlete's current training load is proportionate to their recent conditioning. The formula divides

the past seven days' load by a rolling 28-day average; a ratio close to 1.0 indicates stability, while values above 1.3–1.5 are associated with increased injury risk [5]. In equestrian sport, research has demonstrated that combining heart rate monitoring with exercise load gives a substantially more informative picture of a horse's condition than either metric alone [1]. Horses present an additional complication not present in human sport: gait transitions (walk, trot, canter, gallop) produce very different biomechanical loads, so a single aggregate ACWR score can mask meaningful within-session intensity patterns [4].

Machine learning on tabular sensor data. For tabular data with a few hundred to a few thousand samples, gradient-boosted trees consistently outperform neural networks and are the recommended baseline for this regime [6]. With 820 labeled sessions available, XGBoost [7] is the appropriate choice. Neural network approaches would require substantially more labeled data and would sacrifice the interpretability that tree-based SHAP values provide [6].

Explainable AI with SHAP. SHAP (SHapley Additive exPlanations) [2] assigns each feature a signed contribution to each individual prediction. The critical property for this application is that SHAP is *per-session*: for a given horse on a given day, the system can identify which specific sensor readings pushed the classification in which direction and by how much. This mirrors the structure of expert written feedback, which does not discuss feature importance in general but describes specific measurements for this horse today. SHAP-based explanations have been shown to increase clinician trust in AI decision support [3], and we expected the same dynamic to hold in an equine training context.

Weak supervision for limited labeled data. Obtaining clean labeled data in the equine domain is difficult. The domain expert does not provide pre-labeled datasets; she writes open-ended Dutch text after each session. The weak supervision paradigm [8] addresses this by using data-driven heuristics to assign labels without requiring manual annotation of every session. We used this approach to extend the labeled pool from 300 expert-verified sessions to 820 total, as described in Section II-D.

Multilingual NLP for Dutch-language comments. Where expert comments are available, extracting structured information from Dutch-language text requires a language-agnostic representation. Multilingual sentence transformers [9] produce semantically meaningful embeddings across over 50 languages and provide a foundation for cross-lingual keyword and category extraction without any language-specific fine-tuning. In this study, Dutch keyword extraction uses rule-based regex matching for the labeling phase; the sentence transformer approach represents a direction for richer NLP enrichment in future work.

II. Methodology

This section describes the dataset and preprocessing steps, the module decomposition of the system, the labeling strategy and feature engineering, and the experimental design used to evaluate the prototype.

A. Dataset and Preprocessing

The data comes from a single professional competition yard in the Netherlands and covers the period June to December 2025. The domain expert at this yard has an established research background in equine exercise physiology and workload monitoring [1], meaning the annotations and comments in this dataset reflect a clinically grounded understanding of equine training. There are four source files, summarised in Table I.

TABLE I
Raw Data Sources

File	Contents	Records
eI_ACWR-*.json	Per-horse daily workload metrics	9,771
eI_User_Annotation-*.json	Per-session sensor data	2,743
eI_Communication-*.json	Dutch text comments	2,287
Horses-September2025.xlsx	Horse metadata	62

ACWR file. Records per-horse per-day workload data, including ACWR values broken down by gait [4], cumulative distances, session durations, and an integrated training impulse (iTRIMP) score. A non-obvious property of this file is that several of its fields are themselves JSON strings embedded inside the record—for example, `externalDistance` looks like a flat value but contains a nested JSON object with per-gait breakdowns. Skipping the flattening step means losing approximately 40 important workload features.

Annotation file. Records the rider's post-session assessment: horse feel on a 0–4 scale, horse stress on a 0–4 scale, training type, session intensity, and surface conditions. It also contains an `Analyses` blob with per-gait heart rate histories, an Integrated Training Score (ITS), and an Adapted Training Score (ATS). The combination of these physiological signals with rider perception is consistent with the multi-signal monitoring approach advocated in equine exercise physiology research [1].

Communication file. Contains all text comments from riders and from the domain expert. A critical detail: the domain expert's entries always carry `activity_id = -1`. They are associated with a horse on a given date, not with a specific session, and must therefore be joined on `(horse_id, date)` rather than on session ID. Missing this linkage results in zero expert comments in the merged table.

Preprocessing pipeline. All preprocessing runs in Python with pandas. Column names are normalised to lowercase snake_case, resolving edge cases introduced by the original CamelCase field names. Nested JSON strings in all source files are parsed and flattened, adding approximately 40 columns per record type. Sources are then merged in four left joins: ACWR onto Annotations on `(horse_id, date)`, rider comments by `activity_id`, expert comments on `(horse_id, date)`, and horse metadata by `horse_id`. After removing rows with missing horse IDs and dropping columns with more than 90 % missing values—including the `discipline` field at 96 % null—the cleaned dataset contains 2,476 sessions and 145 columns. Remaining numeric nulls are filled with per-column medians.

B. Module Decomposition

The system is organised into five modules, each responsible for a distinct step in the pipeline. The decomposition is stepwise: each module produces an artefact consumed by the next, and modules are independently replaceable without touching adjacent code.

Data loader and flattener. Reads all four source files, normalises column names, expands nested JSON strings into flat columns, and performs the four-way merge described above.

Feature engineer. Transforms the cleaned master dataset into the 99-feature numeric matrix. Zero null values are guaranteed by the final median imputation step, so downstream modules never need to handle missing data.

Labeler. Implements both labeling phases (Section II-D) and produces the 820-session labeled training set. It keeps the expert-derived and signal-derived labels separate so that each can be evaluated independently if needed.

Classifier. Trains the three XGBoost classifiers [7] using Leave-One-Horse-Out cross-validation. Models are serialised to disk with their label encoders and feature column lists so that inference on new sessions requires only a feature matrix, not a re-training step.

Decision block generator. Takes a single activity ID, retrieves the corresponding feature row, runs all three classifiers, computes SHAP explanations using TreeExplainer [2], applies rule-based open flags (ACWR above critical thresholds, heat conditions, injury), and assembles a structured JSON block for that session.

C. Feature Engineering

We constructed 99 numeric features organised into seven groups, shown in Table II. Every group covers a different dimension of the information the domain expert reads when writing manual feedback.

TABLE II
FEATURE GROUPS (99 TOTAL)

Group	Features	Key examples
Workload	28	dist_total_acwr, itrimp_zone, load_pressure
Quality	14	recovery_score, its_final_score, session_quality
Perception	3	horse_feel, horse_stress, feel_stress_ratio
Context	5	training_type_enc, overall_intensity
Profile	4	age, is_injury_concern, any_injury_flag
Surface	5	is_hot, is_hills, is_hard_surface
Temporal	3	month, session_number, is_new_horse

Workload group. In addition to the standard distance-based ACWR [4], [5], we include a separate iTRIMP-based ACWR (`itrimp_overall_acwr`) and derive three new features from it. The `itrimp_zone` encodes the iTRIMP ACWR into five bins using the same thresholds as the distance ACWR zone, but without the double-counting effect that a binned version of the same signal would introduce. The `itrimp_vs_dist_ratio` is the ratio of iTRIMP ACWR to distance ACWR and captures sessions where heart rate intensity was disproportionately high or low relative to the distance covered—a pattern identified as clinically informative in equine workload research [1]. The `hr_data_quality` flag is 1 when the HR sensor was active and 0 otherwise, allowing the model to discount ITS-derived features in sessions where the sensor was absent. Per-gait ACWR values and 7-day cumulative totals are also included, so the workload group covers both aggregate and gait-specific load.

Quality group. The `session_quality` feature is a composite score computed as ITS $\times$ recovery_score / 10. When the HR sensor is off and ITS is zero, using ITS directly would make the session appear to be of zero quality even though the sensor was simply absent. We address this with a fallback: when ITS = 0, `session_quality` is computed from overall_intensity $\times$ 2 instead, mapping the 1–5 intensity scale to the approximate ITS range (2–10) and distinguishing "no HR data" from "genuinely poor session quality."

Temporal group. Two design decisions are worth noting. We retain `month` as a seasonal signal: summer months correspond to the competition peak for most horses in this yard, while winter is the base conditioning period, and removing `month` reduces macro F1 by approximately 0.023. We exclude `day_of_week` on the grounds that it is not a causal signal—it is a scheduling choice by the rider, not a property of the horse—and its presence in early prototypes caused it to appear in SHAP rankings [2], displacing physiological features. Similarly, the early prototype included an `acwr_zone` feature (a binned version of the distance ACWR), which was removed because it introduced the distance ACWR signal twice and inflated its apparent SHAP importance. The `is_new_horse` flag marks a horse's first three sessions, during which the 28-day chronic window is not yet full and ACWR values are unreliable [4].

D. Labeling Strategy

Obtaining labeled training data was the most difficult engineering challenge in this study. The domain expert did not supply a pre-labeled dataset; she writes open-ended Dutch text after each session. Labels had to be constructed in two phases, following the spirit of the weak supervision paradigm [8].

Phase A — Expert-derived labels (300 sessions, 36.6 %). The domain expert uses a consistent emoji shorthand in her comments: a star indicates the session went well (`on_track`) and a lightning bolt indicates something needs attention (`monitor`). We scanned all of her 303 comments for these patterns and obtained 300 labeled sessions. A word-boundary regex was also applied for Dutch keywords indicating active restrictions (such as *rust* for rest and *herstel* for recovery), using boundary matching to avoid false triggers from partial-word matches such as *rustig* (calm). Where additional semantic classification of comments is needed, multilingual sentence transformers [9] offer a scalable path; for the purposes of this study the keyword approach was sufficient. Phase A labels are treated as the highest-quality part of the training set and assigned a confidence of 0.90, since they come directly from the domain expert's annotations.

Phase B — Signal-derived labels (520 sessions, 63.4 %). For sessions without expert comments, we implemented a multi-signal majority voting algorithm, following the weak supervision approach [8]. Eight signals cast weighted votes for one of three label classes: distance ACWR (weight 2.0),

recovery score (1.5), workload score (1.5), ITS score (1.0), horse feel (1.0), horse stress (1.0), injury flags (3.0), and training type plus intensity context (0.5 each). A session is labeled only when the winning class receives at least 70 % of the total vote weight. Sessions with ambiguous or conflicting signals are left unlabeled rather than being assigned a low-confidence label, following the "abstain rather than guess" principle of the weak supervision literature [8]. The mean confidence of signal-derived labels is 0.809.

The final label distribution over 820 sessions is shown in Table III. One structural limitation is visible: all 102 `recovery_required` labels come from Phase B signal voting. The domain expert's written comments in Phase A produced only `on_track` and `monitor` labels, because she uses the emoji system rather than explicitly writing "recovery required." This means the model's knowledge of the recovery class is learned entirely from sensor thresholds rather than from the expert's direct judgment.

TABLE III
LABEL DISTRIBUTION (820 LABELED SESSIONS)

Class	Count	%	Highest-quality source
on_track	507	61.8	Expert + signal
monitor	211	25.7	Expert + signal
recovery_required	102	12.4	Signal only
Total	820	100	

E. Classifier Design

Three XGBoost classifiers [7] were trained: one for training status (the main classifier), one for risk level, and one for load recommendation. All three share the same 99-feature input. The risk and load labels are derived from status labels: `on_track` maps to `low` risk and `maintain` load; `monitor` maps to `medium` risk and `reduce` load; `recovery_required` maps to `high` risk and `decrease` load. Training them as separate classifiers rather than deriving them at inference time gives each model independent confidence scores and slightly different feature weight distributions.

Hyperparameter settings are the same for all three models: 300 estimators, max depth 4, learning rate 0.05, row subsampling 0.8, and feature subsampling 0.8. These settings were selected to limit overfitting given the dataset size, in line with the recommendation to prefer conservative hyperparameters for gradient-boosted trees in the small-data regime [6].

F. Experimental Design

The main experiment evaluates all three classifiers under Leave-One-Horse-Out (LOHO) cross-validation: for each of the 62 horses, we train on all labeled sessions from the other 61 horses and predict on every labeled session from the held-out horse, then repeat for each horse. This is the strictest possible evaluation for this data structure because it tests generalisation to a horse the model has never encountered during training—the realistic deployment scenario. A standard random 80/20 split would be misleading here: multiple sessions from the same horse appearing in both train and test would inflate performance, and with 62 horses even a 20 % test split would give only 12–13 test horses on average, too few to draw reliable conclusions about per-horse generalisation [6].

III. METRICS

To evaluate the system we use standard metrics from the classification literature, applied consistently across the LOHO folds.

A. Per-Class Precision, Recall, and F1

For each of the three status classes $c \in \{\text{on\_track, monitor, recovery\_required}\}$, let TP_c, FP_c, and FN_c be the true positives, false positives, and false negatives aggregated over all LOHO folds. Precision, recall, and F1 are then:

$$P_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c}, \quad R_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c}, \tag{1}$$

$$F1_c = \frac{2 \cdot P_c \cdot R_c}{P_c + R_c}. \tag{2}$$

Macro F1 averages $F1_c$ equally across the three classes and is the primary summary metric, since it treats each class as equally important regardless of class size—appropriate here given the safety-critical nature of the `recovery_required` class.

B. Overall Accuracy

Overall accuracy is the fraction of sessions classified correctly across all LOHO folds:

$$\text{Acc} = \frac{\sum_c \text{TP}_c}{\text{Total sessions}}. \tag{3}$$

Accuracy is reported alongside macro F1; it is less informative than F1 for imbalanced classes but provides a commonly expected reference point.

C. Per-Horse Macro F1

Because LOHO evaluation produces a separate prediction set for each horse, we can also compute macro F1 per horse and summarise its distribution across all evaluated horses. This exposes the variance that a single pooled aggregate metric conceals, and reveals whether underperformance is concentrated in data-sparse horses.

D. SHAP Feature Importance

For the STATUS classifier, we compute SHAP values using TreeExplainer [2] across all 2,476 sessions. The multi-class XGBoost [7] model produces a three-dimensional SHAP array of shape (sessions × features × classes). Global feature importance is the mean of the absolute values across sessions and classes. In addition to this global ranking, the decision block generator produces a per-session SHAP waterfall for each prediction, naming the top five features and their signed contributions [3].

IV. Results

The prototype was trained and evaluated on 820 labeled sessions from 62 horses. Sixty horses had at least two labeled sessions and were included in the LOHO evaluation; the remaining two had only one labeled session and were excluded since a fold cannot be evaluated with a single test session. Aggregated results are shown in Table IV; per-class results for the STATUS model are shown in Table V; the confusion matrix is in Table VI.

Three headline patterns are visible. First, all three classifiers achieve accuracy above 81 % and macro F1 above 0.79 under LOHO cross-validation—a strong result given the dataset size and the weak-labeling construction of the training set [8]. Second, the `recovery_required` class achieves the highest per-class F1 (0.92) despite being the smallest class: 94 % of the sessions that truly need recovery are correctly identified. Third, the `monitor` class is the weakest (F1 = 0.60): roughly half of the borderline sessions are predicted as `on_track`, which is the hardest classification boundary in the problem.

TABLE IV
LOHO Cross-Validation Results for All Three Classifiers (60 horses evaluated; 820 labeled sessions total).

Model	Accuracy	Macro F1	LOHO Folds
STATUS (main)	0.822	0.800	60
RISK	0.817	0.794	60
LOAD recommendation	0.817	0.794	60

TABLE V
Per-Class Results for the STATUS Model (Pooled LOHO Predictions, $n = 820$ sessions).

Class	Precision	Recall	F1	Support
monitor	0.73	0.52	0.60	211
on_track	0.83	0.93	0.88	507
recovery_required	0.90	0.93	0.92	102
Macro average	0.82	0.79	0.80	820

TABLE VI
STATUS Model Confusion Matrix (LOHO Pooled Predictions). Rows are true classes; columns are predicted classes.

True \ Predicted	monitor	on_track	recovery
monitor	**109**	94	8
on_track	35	**470**	2
recovery_required	6	1	**95**

Table VII shows the top 15 features by mean absolute SHAP value. The distance-based ACWR (`dist_total_acwr`) is the single most important feature, consistent with its established role as the primary training load indicator in sports science [4], [5]. The `session_quality` composite (rank 2) and `its_final_score` (rank 5) together represent the iTRIMP-derived group; their combined SHAP of 0.660 exceeds the distance ACWR alone (0.647), confirming that heart rate training load is at least as informative as distance-based load [1]. This aligns directly with the domain expert's stated view during the project, and provides evidence that the model has learned a signal weighting that mirrors clinical reasoning.

TABLE VII
Top 15 Features by Mean Absolute SHAP Value [2] (STATUS Model, Averaged Across All Sessions and All Three Output Classes).

Rank	Feature	Group	SHAP
1	dist_total_acwr	Workload	0.647
2	session_quality	Quality	0.481
3	recovery_score	Quality	0.298
4	dur_total_acwr	Workload	0.213
5	its_final_score	Quality	0.179
6	session_number	Temporal	0.142
7	intensity_variation	Quality	0.141
8	dur_walk_acwr	Workload	0.137
9	dur_trot_acwr	Workload	0.121
10	month	Temporal	0.118
11	training_spec_enc	Context	0.081
12	dist_canter_slow_7d_total	Workload	0.080
13	dist_trot_7d_total	Workload	0.075
14	load_pressure	Workload	0.073
15	dur_trot_7d_total	Workload	0.068

Per-horse results: of the 60 horses evaluated in LOHO, 36 (60 %) achieved macro F1 $\geq$ 0.75, 21 (35 %) fell in the 0.50–0.75 range, and 3 (5 %) fell below 0.50. The three worst-performing horses each had only one or two labeled sessions in the dataset. Horses with larger labeled sets performed markedly better on average, with a mean macro F1 of 0.797 and a median of 0.822 across the 60 evaluated horses. This variance is consistent with the finding [6] that data volume per group has a direct and non-linear effect on tree-based model performance.

V. Discussion

We discuss the results class by class, then reflect on the engineering approach and its concrete lessons.

A. STATUS Accuracy and Macro F1

The 82.2 % accuracy and 0.800 macro F1 represent a meaningful result, but we want to be precise about what they measure. The training labels consist of 300 expert-derived Phase A labels and 520 signal-derived Phase B labels [8]. Evaluating on Phase B labels means that some fraction of the "correct" answers were generated by the same signal voting algorithm used to construct the training set. A model that accurately reproduces Phase B patterns will score high on those labels regardless of whether they match the domain expert's actual judgment. The Phase A labels—the expert's own emoji annotations—are genuine ground truth and provide the more reliable signal within the LOHO results.

B. Recovery Class Performance

The `recovery_required` class achieves the highest F1 at 0.92, with 95 out of 102 true recovery sessions correctly identified. This is the most practically important class: missing a horse that needs rest can lead to injury or overtraining, a risk highlighted consistently in the workload monitoring literature [4], [5]. The 7 misclassified recovery sessions were all

predicted as monitor rather than on_track—a conservative error that flags a concern rather than dismissing the session as normal.

That said, this class carries the most important caveat in the study: all 102 recovery_required labels are signal-derived. The model's high F1 on this class reflects its ability to reproduce the signal-voting rules used to generate those labels, not necessarily its agreement with the domain expert's clinical judgment. Until expert-verified gold labels are obtained for this class, the high F1 should be treated as a measure of internal consistency rather than confirmed expert alignment.

C. Monitor Class and the Boundary Problem

The monitor class has the lowest F1 at 0.60, driven by a recall of only 52 %. Approximately half of all true monitor sessions are predicted as on_track. This is a structural problem: monitor sessions are borderline by definition, sitting between clearly normal and clearly concerning. Their feature distributions overlap substantially with on_track sessions, and the signals that distinguish them—slightly elevated ACWR, slightly below-average recovery, moderate horse stress—are individually ambiguous [1]. The confusion matrix (Table VI) shows that 94 monitor sessions are misclassified as on_track, compared to only 35 in the reverse direction, suggesting a slight bias toward the majority class.

D. Per-Horse Variation

The LOHO evaluation exposes variance across horses that pooled metrics conceal. The standard deviation of per-horse macro F1 is 0.211, which is substantial. The 3 horses below F1 = 0.50 each had only one or two labeled sessions; in those folds, a single misprediction produces F1 = 0 for that fold. This is a data scarcity problem rather than a model failure. Increasing labeled session count for data-sparse horses—even by two or three sessions—would have a disproportionately large effect on minimum per-horse performance.

E. Feature Relevance and SHAP Interpretability

The combined SHAP weight of the iTRIMP-derived group (session_quality at 0.481 and its_final_score at 0.179, total 0.660) exceeds the single largest feature, dist_total_acwr (0.647). This confirms quantitatively that the domain expert's assertion—that heart rate training load should be the primary signal—is reflected in the learned model weights [1]. The per-gait ACWR values (dur_walk_acwr at rank 8, dur_trot_acwr at rank 9) also contribute information above and beyond the aggregate ACWR, consistent with the argument that gait-level load breakdown matters physiologically for horses [4]. The month feature (rank 10, SHAP 0.118) provides a legitimate seasonal signal; removing it reduces macro F1 by approximately 0.023.

The SHAP explanation outputs [2] serve a dual purpose in this system: as a development tool for catching spurious features, and as a user-facing explanation for each generated block. Two features present in early prototypes were removed following inspection of their SHAP rankings. day_of_week appeared at rank 4 despite having no physiological justification—the day a session occurs is a scheduling decision, not a load signal—and its inclusion was an artefact of spurious correlations in the training labels. acwr_zone, a binned discretisation of dist_total_acwr, doubled the representation of that signal and inflated its apparent importance. The interpretability of SHAP outputs was thus essential not only for explaining predictions to the domain expert [3] but for improving the feature set during development.

F. Was the System of Added Value Here?

At 820 labeled sessions, a well-configured XGBoost model [6], [7] achieves meaningful performance. The added value of the full pipeline, beyond a simple classifier, comes from two design choices. First, SHAP explanations [2] make every prediction inspectable at the session level. Research on clinical AI adoption shows that practitioners are more willing to act on system recommendations when they can verify the reasoning behind them [3], and the same dynamic is expected here: a domain expert reviewing a block can see exactly which sensor readings triggered the recommendation and decide whether to trust or override it. Second, the rule-based open flags fire independently of model confidence and cover hard constraints (extreme ACWR, heat stress, injury) that the model should not be solely responsible for flagging.

The system does not claim to replace the domain expert, and the results do not support that claim. What the results do support is that a structured pipeline can produce a reliable, traceable first draft for the large majority of sessions, leaving the expert to focus on the genuinely ambiguous cases.

G. Engineering Reflection on the Approach

What held up well. The five-module decomposition remained stable throughout the study. Changes to one module—such as adding iTRIMP-derived features to the engineer, or raising the Phase B confidence threshold from 60 % to 70 %—required no changes to adjacent modules. The LOHO evaluation was the right choice: a random split would have hidden the per-horse variance visible in the results and would have overstated the system's generalisation performance [6]. Raising the Phase B confidence threshold from 60 % to 70 % produced a cleaner training set and improved macro F1 on held-out horses by approximately 0.018.

Where the approach fell short. Three things were less satisfying. First, the circular dependency between Phase B labels and the feature set [8] is a fundamental limitation: 63.4 % of training labels were generated using the same sensor signals that appear as features, meaning the model partially learns to reproduce its own training labels. This is mitigated by the Phase A labels but not eliminated. Second, the recovery_required class has no expert-derived labels at all, which means we cannot currently verify that the high F1 on this class corresponds to genuine expert agreement. Third, SHAP computation [2] across all 2,476 sessions is the most compute-intensive pipeline step and currently runs at inference time rather than being cached, which would be a bottleneck at deployment scale.

Concrete recommendations. From the experience of building and evaluating this prototype, we offer the following recommendations for similar sensor-driven decision support systems:

1) *Audit every feature for causal plausibility before including it.* Features like `day_of_week` that correlate with labels but have no causal mechanism will appear important in SHAP [2] and mislead both the model and any expert who inspects the explanations.
2) *Separate label sources and track them throughout.* Keeping expert-derived and signal-derived labels in separate columns makes it possible to evaluate performance on each subset independently, and surfaces the circular dependency described in weak supervision [8] early rather than after training.
3) *Use Leave-One-Group-Out evaluation whenever data has group structure.* In this application, sessions are grouped by horse. Using random splits would have inflated all metrics and masked the per-horse variance that matters most for deployment [6].
4) *Build the SHAP explanation step before presenting results to the domain expert.* Global SHAP rankings [2] catch spurious features early. Session-level SHAP is what the expert needs to evaluate whether the system's reasoning makes sense, and building this transparency in from the start increases the likelihood of adoption [3].

H. Limitations

Four limitations should be stated explicitly. (i) The Phase B training labels were generated from the same sensor signals that appear as features, creating a partial circular dependency [8] that inflates evaluated performance on those labels. The extent to which this affects agreement with the domain expert's actual judgment is an open question. (ii) The `recovery_required` class has no expert-derived labels; its performance metrics reflect signal reproducibility rather than confirmed expert alignment [1]. Collecting even a small set of expert-verified gold labels for this class would be the highest-value data collection effort for a future iteration. (iii) All data originates from a single yard; the feature–label relationships learned by the model are specific to one expert's training philosophy and one discipline mix, and may not transfer to a different yard without re-labeling and re-training [4]. (iv) SHAP computation at deployment scale (streaming sessions from a live application) [2] would require result caching or approximation to remain performant.

VI. Conclusion

This paper proposed and instantiated a machine learning engineering approach for automated training feedback generation in equine sport. The system covers 62 horses and 2,476 sessions, engineers 99 sensor-derived features organised into seven groups, and trains three XGBoost classifiers [7] using a two-phase weak labeling strategy [8] and Leave-One-Horse-Out cross-validation. SHAP-based explanations [2] make every prediction traceable to the specific sensor signals that drove it, satisfying the interpretability requirement that clinical decision support systems face when building trust with domain experts [3].

On all three classifiers the accuracy exceeds 81 % and macro F1 exceeds 0.79 under the strictest possible evaluation. The dominant feature is the distance-based ACWR [4], [5], but the combined SHAP weight of the iTRIMP/ITS group exceeds the ACWR alone, consistent with the equine exercise physiology literature on the importance of heart rate training load [1]. The `recovery_required` class achieves F1 = 0.92, with misclassifications landing conservatively at `monitor` rather than `on_track`. The `monitor` class (F1 = 0.60) remains the open problem, and its resolution will require additional expert-derived Phase A labels or a refined boundary definition for this class.

The gaps are clearly identified. The circular dependency between Phase B labels and the feature set is the most important structural limitation. The most immediate follow-ups are: obtaining expert-verified gold labels for the `recovery_required` class; collecting additional Phase A expert annotations for the borderline monitor sessions to improve model confidence on that class; and evaluating performance on data from a second yard to assess cross-context generalisability [6]. Beyond these steps, the modular pipeline design is intended to remain extensible: new signal groups can be added to the feature engineer independently, and richer Dutch-language NLP components [9] can augment the label quality and comment-based evidence for future iterations.

References

[1] C. C. B. M. Munsters, B. R. M. Kingma, J. van den Broek, and M. M. Sloet van Oldruitenborgh-Oosterbaan, "A prospective cohort study on the acute:chronic workload ratio in relation to injuries in high level eventing horses: A comprehensive 3-year study," *Preventive Veterinary Medicine*, vol. 179, p. 105010, 2020, doi: 10.1016/j.prevetmed.2020.105010.

[2] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[3] S. Tonekaboni *et al.*, "What clinicians want: contextualizing explainable machine learning for clinical end use," *Proceedings of Machine Learning Research*, vol. 106, 2019.

[4] T. J. Gabbett, "The training-injury prevention paradox: should athletes be training smarter and harder?" *British Journal of Sports Medicine*, vol. 50, no. 5, pp. 273–280, 2016.

[5] B. T. Hulin *et al.*, "Spikes in acute:chronic workload ratio associated with a 5–6-fold increase in injury risk in elite rugby league players," *British Journal of Sports Medicine*, vol. 50, no. 4, pp. 231–236, 2016.

[6] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," *Information Fusion*, vol. 81, pp. 84–90, 2022.

[7] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. ACM SIGKDD*, 2016, pp. 785–794.

[8] A. Ratner *et al.*, "Snorkel: Rapid training data creation with weak supervision," *The VLDB Journal*, vol. 29, pp. 709–730, 2020.

[9] N. Reimers and I. Gurevych, "Making monolingual sentence embeddings multilingual using knowledge distillation," in *Proc. EMNLP*, 2020.