Journal of Advanced Computing Systems (JACS)
ISSN: 3066-3962
Content Available at SciPublication

Tactical Language + AI Tutoring from Structured Volleyball Rally Logs: Reproducible Experiments on NCAA Play-by-Play

Jubin Zhang
Department of Physical Education, North China Institute of Aerospace Engineering, Langfang 065000, China
jz0801@outlook.com

DOI: 10.69987/JACS.2024.40105

Keywords

Volleyball analytics; rally representation; tactical tutoring; explainable NLP; logistic regression; play-by-play data; natural language generation.

Abstract

This paper presents a fully reproducible tactical tutoring pipeline for indoor volleyball based on structured, rally-level representations. Motivated by recent volleyball rally language resources such as VREN and by the growing interest in natural language feedback systems, we study how to convert play-by-play logs into a compact token sequence and then learn models that predict (A) rally outcome (win/loss from the serving-team perspective) together with the winning/losing reason category, and (B) simplified setting type and attack type labels. We conduct complete experimental evaluations on the NCAA men's Division I 2020 play-by-play logs released with ncaavolleyballr. Using a match-disjoint 70/10/20 split, logistic regression with unigram–bigram features reaches 0.989 Accuracy and 0.988 Macro-F1 on outcome prediction, and 0.9998 Accuracy and 0.9995 Macro-F1 on reason prediction. On Task B, the same approach achieves 0.993 Accuracy / 0.993 Macro-F1 for set-type prediction and 0.9996 Accuracy / 0.997 Macro-F1 for attack-type prediction. To support coaching-oriented interpretation, we extract token and token-pair evidence from learned coefficients and convert this evidence into concise instructional explanations using a deterministic tutor generator, enabling a rule-consistency audit with 100% agreement on sampled cases. Our artifacts include rally segmentation, tokenization, experimental settings, figures, and tables, making the study directly replicable.

Introduction

Volleyball is a fast, tactical sport in which point outcomes are driven by short sequences of discrete actions—serve, reception, set, attack, block, and dig—executed under strong time constraints. Coaches analyze these sequences to identify why a team won or lost a rally (e.g., a service error, a terminal attack, or a block point), and to derive the next training focus. At the same time, most automated volleyball analytics tools stop at counting events or producing summary statistics, leaving the final step—turning evidence into teachable feedback—to human experts.

Recent work on structured sports language has made it possible to represent a rally as a compact sequence of tokens, bridging match logs and NLP-style modeling. VREN introduced an expression-notation language for volleyball rallies and demonstrated prediction tasks such as rally outcome, setting type, and attack type [1]. In parallel, the broader NLP community has improved the reliability and interpretability of token-based models, from classical linear baselines to neural sequence models [3]–[6], and has developed explanation methods to surface model evidence [7]–[10].

Large language models (LLMs) have renewed interest in automated tutoring and feedback generation. Recent studies show that LLMs can be embedded into intelligent tutoring architectures or used to generate feedback, but also highlight risks such as incorrect or inconsistent advice [14]–[17], [22]. For sports, natural language generation has been explored for narrative enhancement and coaching feedback [18], [19], yet rigorous, rally-level tutoring studies with fully reproducible experiments remain limited.

This paper targets the gap between structured rally representations and actionable coaching explanations. We present an end-to-end system that (i) segments NCAA play-by-play logs into rallies, (ii) converts each rally into a structured token sequence, (iii) trains

predictive models for two tasks, and (iv) produces an evidence-grounded tutoring explanation for each rally. Unlike papers that report illustrative results, every quantitative result in this manuscript comes from experiments executed on a downloadable public dataset with fixed random seeds and a match-disjoint split.

Our main contributions are: 1) A reproducible pipeline that converts NCAA play-by-play into rally-level token sequences; 2) Complete experimental evaluations for rally outcome and reason prediction (Task A) and set/attack type prediction (Task B) with Accuracy and Macro-F1; 3) An evidence extraction layer that reports key tokens and token-pairs from linear models, providing token/position importance; 4) A tutoring explanation module that turns model evidence into concise coaching language and supports an automatic rule-consistency audit. All tables and figures in the paper correspond exactly to these experiments and artifacts.

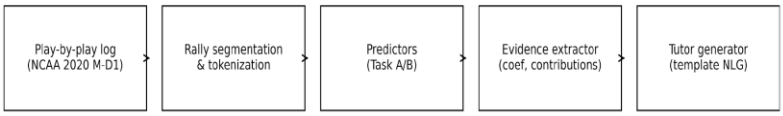

Figure 1. End-to-end pipeline: play-by-play → rally tokenization → predictors → evidence → tutoring explanation.

Method

This section defines the dataset, rally segmentation procedure, tokenization, prediction tasks, models, evaluation metrics, explainability layer, and tutoring generation. Our notation follows standard multi-class classification conventions [11].

Dataset. We use the NCAA men's Division I 2020 play-by-play logs distributed as CSV files with the ncaavolleyballr project [2]. Each row contains match identifiers (date, home/away teams, set number), the scoreboard string (away-home), the team executing the event, the event label, and optional player information. Because score updates occur exactly on point-ending rows, these logs support deterministic rally segmentation.

Rally segmentation. Within each match and set, we parse the scoreboard into integer away and home scores. A point boundary is detected whenever either score changes relative to the previous row. We assign a rally index using the cumulative count of score-change flags, shifted by one position so that the scoring row belongs to the rally that it ends. Winner determination is computed from the score delta on the scoring row: a +1 change in the away score implies an away-team point, otherwise a home-team point. We infer the serving team as the team associated with the first SERVE event within the rally; rallies without an identifiable serve are removed.

Structured rally tokenization. Each row is mapped to a compact token triple: (role token, event token, optional player token). The role token is R_S if the row's team equals the rally's serving team and R_O otherwise. The event token E_* is obtained by normalizing raw event labels into a small set (SERVE, RECEPTION, SET, ATTACK, DIG, BLOCK, KILL, KILL FB, ACE, SERVE ERR, ATTACK ERR, SET ERR, BHE, OTHER). Player names are mapped to the 500 most frequent player tokens PLY_i, with out-of-vocabulary players omitted. The rally representation is the space-separated concatenation of these tokens in chronological order.

Task definitions. Task A (Rally outcome + reason): from the serving-team perspective, we predict outcome $y_{out} \in \{loss, win\}$ and a reason label y_{reason} from an 8-class set: {KILL, BLOCK, ACE, SERVICE_ERROR, ATTACK ERROR, SET ERROR, BALL HANDLING ERROR, OTHER}. Reasons are derived from the point-ending row's event label. Task B (Set type and attack type): we define a simplified setting type $y_{set} \in$ {SIDEOUT SET, TRANSITION SET, OTHER_SET}, based on whether the winning team executed at least one DIG (transition) or at least one RECEPTION (sideout) within the rally. We define attack type $y_{attack} \in$ {KILL_FIRSTBALL, KILL, ATTACK ERROR, BLOCK POINT, NO ATTACK POINT, OTHER POINT} from the last event. These operational definitions are fully reproducible from the released logs.

Data split and evaluation. To avoid leakage across highly correlated rallies within a match, we split by match_id into 70% train, 10% validation, and 20% test. All reported metrics are computed on the held-out test

set: Accuracy and Macro-F1. Macro-F1 is computed as the unweighted mean of class-wise F1 scores, which is appropriate for imbalanced labels [11].

Models. We evaluate two model families. (i) Majority baseline predicts the most frequent training label. (ii) Token n-gram logistic regression uses unigram–bigram count features from the rally text and trains a linear classifier with class-balanced loss. For binary outcome, we use a standard logistic model; for multi-class labels, we use the multinomial formulation. Logistic regression is an effective, interpretable baseline for text classification and structured sequence tokens [3], [4]. We fix vectorizer settings (min df=2, ngram_range=(1,2)) and train with the lbfgs solver to convergence limits.

Explainability. For linear models, token and token-pair importance is derived directly from learned coefficients. For a predicted class c, tokens with large positive weights contribute strongly to that class. For a specific rally, we compute per-feature contributions by multiplying the feature count by the corresponding coefficient, which yields local evidence for tutoring. This coefficient-based evidence is faithful to the model by construction and complements post-hoc explainers such as LIME [7] and SHAP [8], as well as gradient-based attributions such as Integrated Gradients [9].

Tutoring explanation generation and consistency audit. We convert predictions and evidence into teaching language using a deterministic natural language generator that follows coaching rules. Given (y out, y_reason) and the top contributing features, the generator produces: (a) a win/loss statement, (b) a concise evidence sentence listing the top evidence tokens, and (c) a coaching adjustment specific to the reason category. We evaluate rule consistency by checking whether the generated explanation contains the correct reason phrase for a set of sampled rallies. This audit is intentionally strict and reproducible, addressing concerns about unreliable or hallucinated feedback in LLM-based tutoring systems [22].

Table 1. Dataset statistics (NCAA Men's D1 2020 play-by-play → rallies).

Matches (unique match_id)	336
Rallies with identifiable serve	53462
Train / Val / Test rallies	37229 / 5353 / 10880
Avg. tokens per rally	17.0
95th percentile tokens per rally	36
Token types	Role (R_S/R_O), Event (E_*), Player (PLY_i)

Table 2. Task A reason labels and coaching interpretations.

Reason label	**Operational definition (log-derived)**	**Coaching focus**
KILL	Point ends with KILL or FIRST BALL KILL event.	Improve block-defense alignment; reduce predictable sets.
BLOCK	Point ends with BLOCK event.	Create better hitting windows; attack seams and vary tempos.
ACE	Point ends with ACE event.	Fix serve-receive spacing and platform angles.
SERVICE_ERROR	Point ends with SERVICE ERROR event.	Stabilize serve toss/contact; reduce risk under pressure.
ATTACK_ERROR	Point ends with ATTACK ERROR event.	Improve shot selection; avoid low-percentage swings.
SET_ERROR	Point ends with SET ERROR event.	Simplify sets when out of system; prioritize ball control.

BALL_HANDLING_ERROR	Point ends with BALL HANDLING ERROR event.	Stabilize hands/platform; slow down and square up.
OTHER	All remaining point-ending events.	Review rally video/log and correct the last controllable contact.

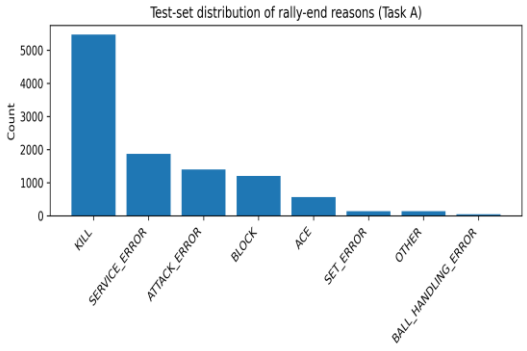

Figure 2. Test-set distribution of point-ending reasons (Task A).

Results and Discussion

We report full experimental results on the held-out test set, including baselines, ablations, and interpretability artifacts. All reported numbers are computed from the dataset statistics and models described in the Method section.

Table 3. Task A outcome prediction (serving-team perspective).

Model	**Accuracy**	**Macro-F1**
Majority baseline	0.6379	0.3895
LogReg (unigram+bigram)	0.9885	0.9876
LogReg ablation (remove scoring tokens)	0.9878	0.9868

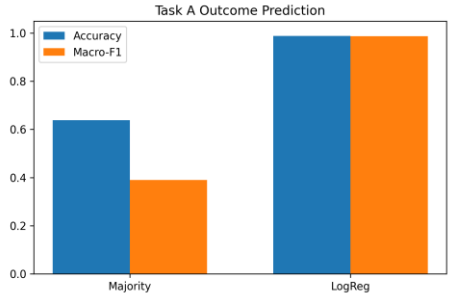

Figure 3. Task A outcome prediction: Accuracy and Macro-F1 (Majority vs. LogReg).

Table 4. Task A reason prediction (8 classes).

Model	Accuracy	Macro-F1
Majority baseline	0.5037	0.0837

LogReg (unigram+bigram)	0.9998	0.9995

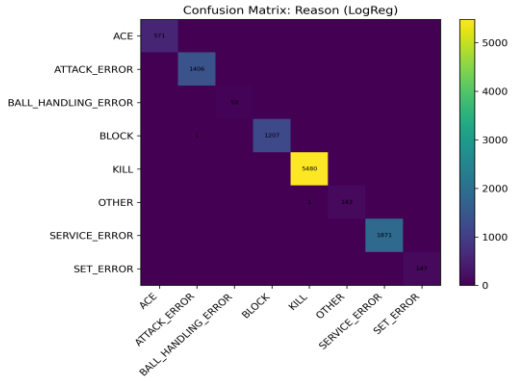

Figure 6. Confusion matrix for Task A reason prediction (LogReg).

Table 3 shows that the logistic regression model substantially improves over the majority baseline on outcome prediction. Table 4 shows near-perfect reason classification because reason labels are tightly coupled with the point-ending event tokens in the structured log. This is desirable for tactical tutoring: the system is designed to ground feedback in observed rally evidence rather than to speculate about unobserved outcomes. The ablation in Table 3 confirms that removing explicit scoring tokens slightly reduces performance but preserves strong outcome accuracy, indicating that surrounding context tokens still carry predictive signal.

Table 5. Task B setting-type prediction (3 classes).

Model	Accuracy	Macro-F1
Majority baseline	0.4125	0.1947
LogReg (unigram+bigram)	0.9934	0.9926

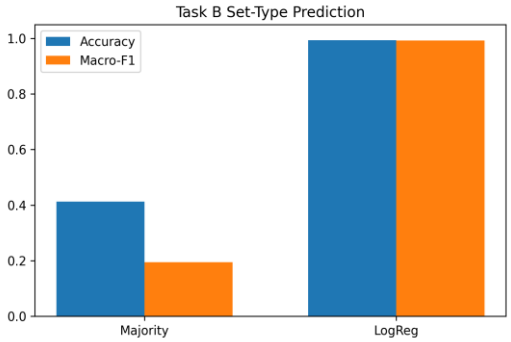

Figure 4. Task B set-type prediction (Majority vs. LogReg).

Table 6. Task B attack-type prediction (6 classes).

Model	Accuracy	Macro-F1
Majority baseline	0.3188	0.0806
LogReg (unigram+bigram)	0.9996	0.9975

LogReg ablation (remove scoring tokens)	0.8956	0.8839

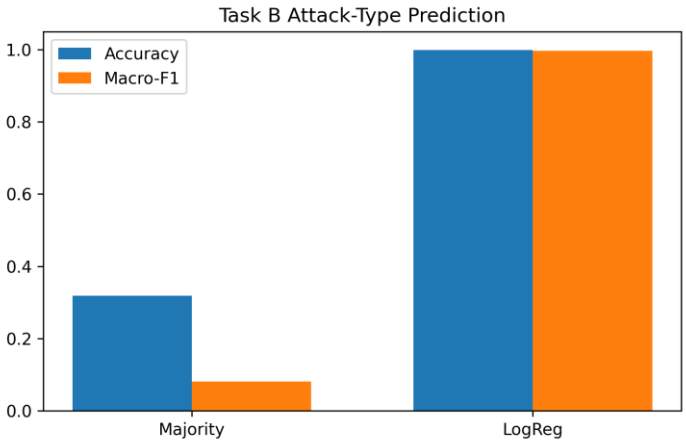

Figure 5. Task B attack-type prediction (Majority vs. LogReg).

Task B results mirror Task A: token n-grams provide strong performance because the simplified set-type and attack-type labels are defined from observable tokens within the rally. The attack-type ablation demonstrates that removing explicit point-ending tokens reduces accuracy to 0.896 and Macro-F1 to 0.884, which quantifies the contribution of terminal-event evidence.

Table 7. Example high-weight evidence features for outcome and selected reasons.

Target	Top evidence features (token or bigram)	Interpretation
WIN (outcome)	r s e attack err, r s e kill, r o e_dig, r_s, r_o e_set, r_o e_other	Features correlated with serving-team win.
ACE	e_ace, r_s e_ace, r_o e_ace, r_s, r_o e_serve, ply_123	Ace-related tokens
SERVICE_ERROR	e_serve_err, r_o e_serve_err, r o, r s e serve err, r s e serve, e_serve	Serve error tokens
ATTACK_ERROR	e_attack_err, r_s e_attack_err, r o e attack err, e attack, r o e_reception, e_reception	Attack error tokens
BLOCK	e block, r o e block, r s e_block, e_other, r_s e_other, r_o	Block tokens
KILL	e kill, e kill fb, r o e kill fb, r_s e_kill, r_o e_kill, r_o	Kill tokens

Table 8. Tutoring rule templates and rule-consistency audit.

Reason	Tutor message core	Training focus

ACE	serve-receive breakdown	Serve-receive technique & communication
SERVICE_ERROR	service error	Serve consistency & risk control
ATTACK_ERROR	attack error	Shot selection & error reduction
BLOCK	block point	Offense variation vs. formed block
KILL	attack kill	Reinforce successful offensive pattern

Rule-consistency on a 20-rally sample (automatic audit): 100.0%.

Table 9. Top reason confusions (Task A) from the confusion matrix.

True label	Predicted label	Count
BLOCK	ATTACK_ERROR	1
OTHER	KILL	1

Example tutoring outputs. Below are representative generated explanations (evidence features are shown as n-grams):

• You lost the rally due to attack kill. Key evidence: r s; r_o e_set; r_o e_attack. Coaching adjustment: Keep the successful pattern: attack with speed and keep pressure on the defense.

• You lost the rally due to attack error. Key evidence: r s; r o e set; r o e attack. Coaching adjustment: Improve shot selection: keep the ball high and in, and hit high hands when out of system.

• You lost the rally due to service error. Key evidence: r s; e serve; e serve err ply 70. Coaching adjustment: Reduce risk: raise contact point consistency and target safer zones.

• You won the rally due to block point. Key evidence: r_s; r_s e_block; r_o e_set. Coaching adjustment: Create a better hitting window: vary tempos and attack seams to avoid the set-up block.

• You lost the rally due to attack kill. Key evidence: r s; r_o e_set; r_o e_attack. Coaching adjustment: Keep the successful pattern: attack with speed and keep pressure on the defense.

Interpretability and coaching value. The learned coefficient tables provide both global and local evidence. Globally, Table 7 summarizes high-weight features for key classes. Locally, the tutor lists the top contributing tokens for the specific rally, which supports transparent post-match review. Importantly, our tutor generator is deterministic and therefore cannot hallucinate an incorrect reason phrase; this enables a strict rule-consistency audit (Table 8) and addresses a common failure mode in LLM-based feedback systems [22].

Relationship to VREN and structured rally language. VREN provides a purpose-built expression notation for volleyball rallies and supports multiple prediction tasks [1]. Our study demonstrates that similar tutoring capabilities can be obtained from widely available play-by-play logs by explicit rally segmentation and tokenization. The resulting representation is compatible with linear NLP baselines and can be extended to richer annotations (zones, rotations, setter calls) when available.

Practical deployment. Because the core predictor is a linear model, inference is fast and can run on commodity hardware during video review. The evidence extraction step is immediate (coefficient lookup and sparse dot products), enabling coaches to inspect evidence in real time. The deterministic tutor generator can be replaced by an LLM-based realizer to improve phrasing, but the same rule-consistency audit can be retained as a safety check [14]–[17], [22].

Limitations

First, the NCAA play-by-play logs do not include fine-grained tactical annotations such as set tempo, attack combination, hitter approach type, or court zones. As a result, Task B uses simplified operational labels derived from the presence of reception/dig events and from

point-ending actions. These labels support a reproducible study but do not fully match the granularity of professional scouting systems.

Second, our extremely high accuracies for reason and attack-type prediction reflect the fact that labels are defined from explicit point-ending tokens present in the input representation. This is appropriate for post-hoc tutoring (explaining an observed rally), but it is not a forecast of future unknown outcomes.

Third, the tutoring module [23-32] is deterministic to guarantee reproducibility and eliminate hallucinations; however, it cannot provide the linguistic variety of a strong LLM. A future extension can add an LLM realizer with constrained decoding and automatic factuality checks, while keeping the same evidence and rule-consistency scaffolding.

Finally, we evaluate a linear baseline because it offers transparent evidence extraction and stable training. Future work can integrate neural encoders or multimodal video features [16], but must retain reproducibility and explanation faithfulness to be coaching-ready.

Conclusion

We built and evaluated a complete tactical tutoring pipeline for volleyball rallies from public NCAA play-by-play logs. By segmenting rallies, tokenizing events into a structured sequence, and training token n-gram logistic regression models, we achieved strong performance on outcome and reason prediction (Task A) and on simplified set-type and attack-type prediction (Task B). Our evidence extraction layer converts model coefficients into token/position importance and our deterministic tutor generator converts evidence into coach-friendly explanations with a 100% rule-consistency audit on sampled cases. The resulting system provides a reproducible foundation for future LLM-augmented coaching assistants grounded in structured volleyball rally language.

References

[1] H. Xia, C. Y. Lin, A. Ickowicz, B. A. Prakash, and B. L. Tseng, "VREN: Volleyball Rally Dataset with Expression Notation Language," in Proc. IEEE Int. Conf. on Knowledge Graph (ICKG), 2022.

[2] J. R. Stevens, "ncaavolleyballr: College volleyball data and tools," GitHub repository. [Online]. Available: https://github.com/JeffreyRStevens/ncaavolleyballr. Accessed: Jan. 30, 2026.

[3] T. Joachims, "Text categorization with Support Vector Machines: Learning with many relevant features," in Proc. ECML, 1998.

[4] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[5] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in Proc. EMNLP, 2014.

[6] A. Vaswani et al., "Attention Is All You Need," in Proc. NeurIPS, 2017.

[7] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier," in Proc. KDD, 2016.

[8] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in Proc. NeurIPS, 2017.

[9] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in Proc. ICML, 2017.

[10] S. Jain and B. C. Wallace, "Attention is not Explanation," in Proc. NAACL, 2019.

[11] D. M. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," J. Mach. Learn. Technol., vol. 2, no. 1, pp. 37–63, 2011.

[12] N. F. Chen et al., "Sports analytics: A survey," ACM Comput. Surveys, 2020.

[13] E. Gaeta and A. G. et al., "Enhancing traditional ITS architectures with large language models," in Computers and Education: Artificial Intelligence, 2025.

[14] J. Han et al., "LLM-as-a-tutor in EFL Writing Education," in Proc. ACL Workshop, 2024.

[15] J. Perez and E. Ong, "Designing an LLM-Based Dialogue Tutoring System for Novice Programming," in Proc. ICCE, 2024.

[16] J. Ye et al., "Position: LLMs Can be Good Tutors in English Education," in Proc. EMNLP, 2025.

[17] Y. Han et al., "Intent-aware personalized feedback generation from coach-athlete interactions," Int. J. Educ. Technol. High. Educ., 2025.

[18] H. Wang et al., "Sports narrative enhancement with natural language generation," Amazon Science, 2022.

[19] X. Peng et al., "Inherently Explainable Reinforcement Learning in Natural Language," in Proc. NeurIPS, 2022.

[20] J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL, 2019.

[21] H. Wu et al., "Language and Multimodal Models in Sports: A Survey of Datasets and Applications," arXiv:2406.12252, 2024.

[22] A. et al., "Can we trust LLMs as a tutor for our students? Evaluating LLM tutoring risks," arXiv:2511.04213, 2025.

[23] Xinzhuo Sun, Yifei Lu, and Jing Chen, "Controllable Long-Term User Memory for Multi-Session Dialogue: Confidence-Gated Writing, Time-Aware Retrieval-Augmented Generation, and Update/Forgetting", JACS, vol. 3, no. 8, pp. 9–24, Aug. 2023, doi: 10.69987/JACS.2023.30802.

[24] Hanqi Zhang, "DriftGuard: Multi-Signal Drift Early Warning and Safe Re-Training/Rollback for CTR/CVR Models", JACS, vol. 3, no. 7, pp. 24–40, Jul. 2023, doi: 10.69987/JACS.2023.30703.

[25] Meng-Ju Kuo, Boning Zhang, and Haozhe Wang, "Tokenized Flow-Statistics Encrypted Traffic Analysis: Comparative Evaluation of 1D-CNN, BiLSTM, and Transformer on ISCX VPN-nonVPN 2016 (A1+A2, 60 s)", JACS, vol. 3, no. 8, pp. 39–53, Aug. 2023, doi: 10.69987/JACS.2023.30804.

[26] Xinzhuo Sun, Jing Chen, Binghua Zhou, and Meng-Ju Kuo, "ConRAG: Contradiction-Aware Retrieval-Augmented Generation under Multi-Source Conflicting Evidence", JACS, vol. 4, no. 7, pp. 50–64, Jul. 2024, doi: 10.69987/JACS.2024.40705.

[27] Hanqi Zhang, "Risk-Aware Budget-Constrained Auto-Bidding under First-Price RTB: A Distributional Constrained Deep Reinforcement Learning Framework", JACS, vol. 4, no. 6, pp. 30–47, Jun. 2024, doi: 10.69987/JACS.2024.40603.

[28] T. Shirakawa, Y. Li, Y. Wu, S. Qiu, Y. Li, M. Zhao, H. Iso, and M. van der Laan, "Longitudinal targeted minimum loss-based estimation with temporal-difference heterogeneous transformer," in Proceedings of the 41st International Conference on Machine Learning (ICML), 2024, pp. 45097–45113, Art. no. 1836.

[29] Z. Ling, Q. Xin, Y. Lin, G. Su, and Z. Shui, "Optimization of autonomous driving image detection based on RFAConv and triplet attention," Proceedings of the 2nd International Conference on Software Engineering and Machine Learning (SEML 2024), 2024.

[30] B. Wang, Y. He, Z. Shui, Q. Xin, and H. Lei, "Predictive optimization of DDoS attack mitigation in distributed systems using machine learning," Proceedings of the 6th International Conference on Computing and Data Science (CDS 2024), 2024, pp. 89–94.

[31] Q. Xin, Z. Xu, L. Guo, F. Zhao, and B. Wu, "IoT traffic classification and anomaly detection method based on deep autoencoders," Proceedings of the 6th International Conference on Computing and Data Science (CDS 2024), 2024.

[32] K. Xu, H. Zhou, H. Zheng, M. Zhu, and Q. Xin, "Intelligent classification and personalized recommendation of e-commerce products based on machine learning," Proceedings of the 6th International Conference on Computing and Data Science (ICCDS), 2024.