



DALS-IDS: DRIFT-AWARE LIGHTWEIGHT STACKING INTRUSION DETECTION FOR SMART TRANSPORT SYSTEMS

Ms.Vhandkar P.S¹, Gayatri Bhosale²

¹ Faculty Guide & Supervisor | ² Dattakala Group of Institutions, India

Corresponding author: gayatribhosale2007@gmail.com

ARTICLE ID IJREK-2026-00168	ARTICLE TYPE Research Article	RESEARCH AREA Computer Science	PUBLICATION Volume 1, Issue Issue 1 (2026)
RECEIVED 26 October 2026	REVISED 27 October 2026	ACCEPTED 26 October 2026	PUBLISHED 28 October 2026

Abstract

Smart transport systems combine in-vehicle Controller Area Network (CAN) communications, roadside infrastructure, and connected network services. These components can be exposed to changing traffic patterns and intrusion attempts, making a detector trained on a fixed dataset potentially unreliable over time. This paper proposes DALS-IDS (Drift-Aware Lightweight Stacking Intrusion Detection System), a research framework for intrusion detection across smart-transport and supporting network environments. DALS-IDS combines LightGBM, ExtraTrees, and Random Forest base learners with feature subsampling and a Logistic Regression meta-learner trained using out-of-fold predictions. To address changing data distributions, a sliding-window performance monitor updates ensemble weights, while ADWIN is used as a concept-drift detector to trigger weight recalculation or recent-window meta-learner retraining. A SHAP-based explanation layer reports influential features for each alert. The design emphasizes leakage-free preprocessing, CPU-only efficiency measurement, per-class evaluation, temporal testing, and cross-dataset generalization. Planned evaluation includes CAN-bus datasets and network intrusion datasets, subject to verification of dataset availability, schema compatibility, and permissible feature mapping. This paper presents a proposal and planned evaluation methodology; it does not claim experimental results or validated performance improvements.

Keywords: intrusion detection; smart transport systems; CAN bus; concept drift; ADWIN; stacking; LightGBM; ExtraTrees; SHAP; edge computing

I. INTRODUCTION

Connected and intelligent transport systems depend on communication between electronic control units (ECUs), invehicle networks, roadside units (RSUs), gateways, and cloud services. Intrusion detection systems (IDSs) can help identify abnormal or malicious traffic, including denial-of-service, fuzzing, and message-spoofing activity. The supplied literature includes work on cloud-based ITS intrusion detection [1] and the wider connected smart-transport environment [2]. The planned evaluation will also consider vehicular CAN-bus and network intrusion datasets identified in the project proposal; their availability, labels, and feature compatibility will be verified before experiments are reported. A practical detector for smart transport must address more than classification accuracy. Attack prevalence may be highly imbalanced, traffic patterns can change with time, and edge deployment places constraints on inference latency and memory. A model evaluated using a random split may also appear more generalizable than it is if records from closely related time periods occur in both training and testing data. Preprocessing operations such as scaling, encoding, and synthetic oversampling can introduce information leakage if they are fitted before the validation split. This paper proposes DALS-IDS, a drift-aware lightweight stacking framework for smart transport intrusion detection. Three tree-based learners—LightGBM, ExtraTrees, and Random Forest—are used as base models. Feature subsampling encourages diversity, and a Logistic Regression meta-learner is trained on out-of-fold (OOF) predictions. A sliding window estimates recent model performance; ADWIN detects changes in a monitored error stream and triggers weight recalculation or meta-learner retraining. SHAP explanations are intended to identify the features that contributed most to an individual alert. The work is a system-design proposal. Its contribution is a reproducible evaluation plan that combines leakage control, drift-aware ensemble adaptation, cross-dataset tests, interpretable alerts, and CPU-only efficiency measurements. No experimental improvement is claimed before implementation and testing.

II. PROBLEM STATEMENT

Many IDS models are trained and assessed under a fixed data distribution. In smart transport, however, normal CAN traffic, network services, attack methods, and operating conditions may vary over time or between vehicles. A detector that performs well on a source dataset may degrade when deployed on a different vehicle, a later time period, or a different network environment. The research problem is to design a computationally practical IDS that combines diverse lightweight classifiers, adapts its ensemble decision process when monitored performance changes, and can be evaluated without preprocessing leakage. The framework must also make its alerts interpretable and measure whether its performance is suitable for CPU-only edge deployment. The central research question is: Can drift-aware weighting and lightweight stacking improve the stability and crossdataset usefulness of intrusion detection in smart transport, while maintaining acceptable inference cost and interpretable alerts? This remains an empirical question and will be answered only after the proposed experiments are completed.

III. OBJECTIVES

The objectives of DALS-IDS are: 1) To design a stacking IDS using LightGBM, ExtraTrees, and Random Forest base learners. 2) To increase model diversity through separate feature subsets for the base learners. 3) To prevent data leakage by fitting preprocessing and SMOTE only within training folds. 4) To use out-of-fold predictions to train a Logistic Regression meta-learner. 5) To monitor recent performance with a sliding window and use ADWIN to detect potential concept drift. 6) To recalculate ensemble weights or retrain the meta-learner when drift is detected. 7) To evaluate per-class recall, macro-F1, PR-AUC, Matthews correlation coefficient (MCC), and false-positive rate. 8) To measure CPU-only latency, throughput, model size, and peak memory use. 9) To generate feature-level explanations for alerts using SHAP. 10) To examine temporal and cross-dataset generalization on network and vehicular datasets.

IV. LITERATURE REVIEW

A. Vehicular CAN-Bus and Network Intrusion Datasets The base paper FUSE-Net studies intrusion detection for cloudbased intelligent transport systems and discusses network datasets used to evaluate ensemble learning [1]. The proposed DALs-IDS extends the evaluation scope to include vehicular CAN-bus data as well as network intrusion datasets. The planned vehicular datasets are Car-Hacking and CICIoV2024; the planned network datasets are CICIDS2017, CSE-CICIDS2018, and UNSW-NB15. These dataset names define the intended evaluation plan, not a claim that the datasets have already been downloaded, harmonized, or experimentally evaluated. The datasets may differ in capture conditions, feature definitions, class labels, and attack distributions. Therefore, cross-dataset evaluation will not assume that matching column names establish semantic equivalence. Before testing, the study will document dataset versions, feature mappings, label mappings, preprocessing decisions, and excluded features. If a defensible common feature space cannot be established, the corresponding transfer experiment will be reported as infeasible rather than forcing an invalid mapping.

B. Ensemble Learning and Feature Subspacing FUSE-Net is the principal technical reference for DALs-IDS because it combines feature subspacing, classifier weighting, and meta-learning for intrusion detection in cloud-based intelligent transport systems [1]. DALs-IDS retains this general ensemble motivation but proposes LightGBM, ExtraTrees, and Random Forest as base learners, with Logistic Regression trained on out-of-fold predictions. Feature subspacing will provide each base learner with a different feature view. Whether this improves diversity and generalization will be assessed experimentally rather than assumed.

C. Drift Awareness and Explainability DALs-IDS proposes a sliding window to estimate recent predictive performance and ADWIN to detect changes in a monitored stream. When a drift alarm occurs, the system will recalculate model weights and, where enough recent labeled data are available, consider retraining the meta-learner. This is a proposed extension to the ensemble approach discussed in FUSE-Net [1]; the behavior and value of drift-triggered updates remain to be validated. A SHAP-based explanation layer is planned to report influential features for each alert. Explanations will be treated as model-attribution evidence, not proof of causation or a guarantee that an alert is correct. The explanation layer and its runtime overhead will be measured as part of the planned evaluation.

D. Base Research: FUSE-Net The primary reference for this proposal is FUSE-Net, a hybrid ensemble framework for intrusion detection in cloud-based intelligent transport systems. It combines feature subspacing, classifier weighting, and meta-learning, using Random Forest, XGBoost, Gradient Boosting, and Support Vector Machines. Its reported evaluation uses network intrusion datasets [1]. DALsIDS adopts the broad motivation of combining diverse learners for transport-related cybersecurity, but proposes LightGBM, ExtraTrees, and Random Forest as base learners and makes drift response, fold-local preprocessing, cross-dataset protocol design, and CPU-only efficiency explicit research targets. Unlike the reported FUSE-Net results, DALs-IDS is a proposal and does not claim measured performance.

E. Smart Transport and Intelligent Infrastructure Context Ahmad et al. describe Industry 4.0 foundations for smart transport, including cyber-physical systems, Industrial IoT sensing, edge-cloud analytics, and secure interoperability [2]. These themes motivate a distributed DALs-IDS design in which time-sensitive inference may run at a vehicle gateway or roadside unit while model management and retraining can be handled in the cloud. Shaaban and Adalbib discuss integrated and sustainable transport in Qatar, including the role of Intelligent Transport Systems in coordinating transport infrastructure and operations [3]. Alqahtani's GlassSkyDeliver study examines an automated last-mile transport concept relying on real-time routing, tracking, and digital verification [4]. Although these works do not propose the DALs-IDS intrusion-detection method, they illustrate connected, data-dependent transport environments in which availability, trustworthy communication, and timely anomaly detection matter. Gao investigates forecast-driven coordination of renewable energy hubs and transport-based storage under economic and social-welfare constraints [5]. This work broadens the smart-mobility context to include connected energy and transport infrastructure. DALs-IDS does not optimize energy dispatch; rather, it addresses the cybersecurity monitoring layer that may support the reliability of interconnected transport and supporting digital services. These contextual papers motivate the deployment setting, while the technical IDS design is primarily grounded in FUSE-Net [1].

V. RESEARCH GAP

The reviewed resources support vehicle-focused and general network IDS evaluation, tree-based learning, stacking, drift detection, resampling, and explainability. However, a single evaluation design must still address several connected issues: dynamic adaptation must be operational rather than merely described as dynamic; training-time preprocessing must be leakage-free; cross-dataset testing must account for schema differences; and performance must include both class-sensitive detection metrics and edge-resource costs. DALs-IDS proposes to combine these elements in one framework. Its intended contribution is not a claim that any individual algorithm is new, but a reproducible design for studying their integration under temporal drift, dataset shift, and edge-deployment constraints.

VI. PROPOSED DALs-IDS ARCHITECTURE

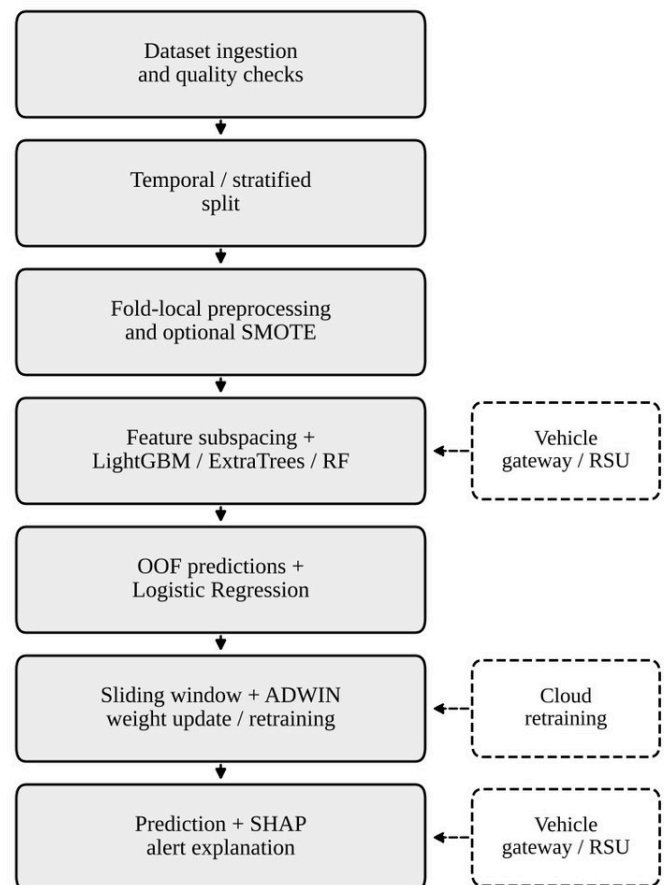


Fig. 1. Proposed DALs-IDS workflow. The diagram represents a design

VII. DATASETS AND CROSS-DATASET DESIGN

TABLE I PLANNED DATASETS AND INTENDED ROLE

Dataset	Planned use	Key verification requirement
Car-Hacking (HCRL)	Source dataset for CAN- bus training; DoS, fuzzy, and spoofing classes	Confirm file versions, fields, labels, and split availability
CICIoV2024	Vehicular cross-dataset test	Map CAN ID, DLC, data bytes, and compatible labels
CICIDS2017	Network-flow training source	Record selected flow schema and class mapping
CSE-CIC-IDS2018	Network cross-dataset test	Establish semantically compatible common features
UNSW-NB15	Additional network transfer test	Align feature definitions and compatible class labels

VIII. PLANNED EVALUATION METHODOLOGY

A. Experimental Protocol The evaluation will use two complementary protocols. First, within-dataset performance will be estimated using stratified five-fold cross-validation where samples can reasonably be treated as independent. Mean and standard deviation will be reported across folds. Second, temporal and cross-dataset experiments will use separate source and target partitions, with no target-test labels used for tuning, resampling, feature selection, or threshold selection. Time-based splits will place older records in training and newer records in testing. The primary comparison will include DALs-IDS, plain stacking without drift adaptation, a static F1-weighted softvoting baseline, standalone Random Forest, standalone LightGBM, and XGBoost. ExtraTrees will also be evaluated as a standalone learner. A CNN or LSTM may be added as an optional comparison if its data requirements and computational cost can be supported. All comparisons will use consistent data partitions and documented tuning budgets.

B. Detection Metrics The planned detection metrics are:

- Macro-F1: gives equal weight to each class and highlights minority-class performance.
- Per-class recall: measures the proportion of each class correctly detected.
- PR-AUC: summarizes precision-recall behavior, especially where positive classes are rare; the positive-class definition will be stated for each experiment.
- MCC: provides a balanced summary of binary classification quality and will be computed using an appropriate multiclass extension when required.
- False-positive rate: measures the proportion of benign examples incorrectly flagged as attacks.

C. Efficiency Metrics Efficiency will be measured on a documented CPU-only machine with hardware, operating system, software versions, thread settings, and batch size reported. The planned measures are single-sample and 1,000-sample inference latency, throughput in samples per second, serialized model size, and peak RAM usage in megabytes. Warm-up, preprocessing time, and model-only inference time will be reported separately where practical. No latency or memory values are claimed in this proposal.

D. Statistical Analysis For cross-validation experiments, results will be reported as mean $\pm$ standard deviation. Where paired test-fold or repeatedrun results are appropriate, the Wilcoxon signed-rank test will be used to compare DALs-IDS with selected baselines. The comparison unit, number of paired observations, significance threshold, and any correction for multiple comparisons will be stated before analysis. Statistical significance will not be interpreted as practical importance without considering effect size and resource cost.

E. Drift Simulation Drift will be evaluated using time-ordered data and controlled changes in the evaluated stream where justified. ADWIN will monitor a preselected error signal when labels are available. The study will report detection delay, false drift alarms, the number of updates, and macro-F1 or error rate before and after drift events. Artificial drift injection will be clearly distinguished from naturally occurring temporal changes in a dataset.

F. Ablation Study The following ablations are planned: 1) remove the drift-aware module and retain plain stacking; 2) remove SHAP explanations to measure their computational overhead, not their effect on predictive accuracy unless the explanation layer influences the pipeline; 3) remove feature subsampling and provide the full feature set to each base learner; 4) compare SMOTE with class weighting and no resampling; 5) compare dynamic weighting alone with dynamic weighting plus recent-window meta-learner retraining.

IX. EXPECTED CONTRIBUTIONS

The expected contributions of DALs-IDS are:

- a leakage-controlled and reproducible training and evaluation pipeline;
- a practical design for updating ensemble weights in response to ADWIN drift alarms;
- a lightweight tree-based stacking configuration intended for CPU-only edge inference;
- a cross-dataset evaluation plan covering vehicular CAN traffic and network flows;
- per-class and class-imbalance-sensitive reporting instead of accuracy alone;
- SHAP-based feature explanations to support inspection of individual alerts;
- explicit measurement of latency, throughput, model size, and peak memory.

These are expected design contributions, not experimentally established outcomes. The proposed framework may fail to outperform simpler baselines, particularly when the datasets are small, feature alignment is weak, or drift labels arrive too late for useful adaptation.

X. LIMITATIONS AND THREATS TO VALIDITY

First, this paper presents a proposal and contains no experimental validation. The effectiveness of dynamic weighting, stacking, and feature subsampling remains to be tested. Second, ADWIN requires a monitored signal and meaningful feedback.

Delayed labels or inaccurate proxy labels may postpone or misdirect adaptation. Third, publicly available CAN datasets can reflect laboratory or controlled capture conditions that differ from live traffic. Fourth, SMOTE may generate unrealistic examples for discrete or structured CAN data; alternatives such as class weighting must therefore be tested. Fifth, cross-dataset evaluation depends on semantically compatible feature definitions and labels. A shared column name does not establish that two features measure the same quantity. Sixth, the proposal does not currently cover adversarial evasion, zero-day detection, or attacks outside the selected label space. Seventh, SHAP explanations may be costly and should be profiled on edge hardware. Finally, performance on a CPU test machine does not by itself establish real-time safety or suitability for a production vehicle. Real deployment would require security, privacy, reliability, and operational testing.

XI. FUTURE WORK

Future work will implement a small prototype and report measured results for detection, drift response, explainability, and resource consumption. Additional directions include online learning, federated learning that avoids transferring raw vehicle data, and unsupervised or semi-supervised anomaly detection for previously unseen attacks. Adversarial robustness testing will be added to examine whether carefully modified traffic can evade the detector. A further step is deployment on a vehicle gateway or RSU tested under controlled conditions. Data collection that reflects a wider range of traffic environments, including African traffic and road conditions, could help investigate geographic and operational variation. Model compression methods such as quantization and pruning may also be studied to reduce the resource requirements of smaller edge devices. These directions will be pursued only after the core evaluation pipeline is reproducible.

XII. CONCLUSION

This paper proposed DALs-IDS, a Drift-Aware Lightweight Stacking IDS for Smart Transport Systems. The framework combines LightGBM, ExtraTrees, and Random Forest base learners, feature subsampling, an OOF-trained Logistic Regression meta-learner, sliding-window performance monitoring, ADWIN-triggered updates, and SHAP-based alert explanations. Its evaluation plan emphasizes leakage-free preprocessing, time-based testing, cross-dataset transfer, perclass detection metrics, and CPU-only efficiency measurements. The proposed study will examine vehicular CAN-bus datasets alongside network intrusion datasets, while explicitly accounting for differences in schema and label definitions. The design also recognizes practical constraints, including delayed feedback for drift detection, the uncertain suitability of SMOTE for structured CAN traffic, and the gap between public datasets and live deployment. DALs-IDS is therefore a research proposal rather than a validated IDS. Its effectiveness and edge suitability can be established only through implementation and controlled evaluation.

$$w_{m,t} = \frac{\exp(\tau s_{m,t})}{\sum_j \exp(\tau s_{j,t})}$$

Vi. Proposed Dals-Ids Architecture

REFERENCES

- [1] V. Govindarajan, M. J. Gnanadhas, A. Wairagade, M. Ayaz, M. Hijji, and U. D. Maiwada, "FUSE-Net: A hybrid ensemble learning framework for securing cloud-based intelligent transport systems in African smart mobility networks," *Scientific African*, vol. 31, Art. no. e03149, 2026, doi: 10.1016/j.sciaf.2025.e03149.
- [2] W. Ahmad, S. Khan, T. Mazhar, T. Shahzad, W. Jiang, and H. Hamam, "Industry 4.0 for smart transport systems: Foundations and applications," *Telematics and Informatics Reports*, vol. 20, Art. no. 100255, 2025, doi: 10.1016/j.teler.2025.100255.
- [3] K. Shaaban and M. A. Adalbib, "Smart and sustainable integration strategies within Qatar's urban transport system," *Procedia Computer Science*, vol. 280, pp. 213–219, 2026.
- [4] A. S. Alqahtani, "GlassSkyDeliver: Design and evaluation of a smart, sustainable, and automated urban transport system for last-mile delivery," *Transportation Research Procedia*, vol. 96, pp. 308–315, 2026, doi: 10.1016/j.trpro.2026.03.040.
- [5] F. Gao, "Forecast-driven economic co-transition of renewable-rich urban energy hubs with eco-friendly transport-based storage systems under political-social welfare in future-ready smart cities," *Renewable Energy*, vol. 269, Art. no. 125796, 2026, doi: 10.1016/j.renene.2026.125796.

Publication archival note: Official scholarly contribution published under CC BY 4.0 International License. Permanent record on <https://www.ijrek.org> | Article ID: IJREK-2026-00168