ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 10, Issue 6, June 2025

A Tree-Based Multi-Tiered Decision Ensemble Framework for Intelligent Intrusion Detection in the Internet of Vehicles

[1] Achatha R, [2] Greeshma O, [3] Manisha S, [4] Sindhu M, [5] Dr.N.Shunmuga Karpagam
[1] [2] [3] [4] Department of CSE, Er.Perumal Manimekalai College of Engineering, Hosur
[5]Associate Professor, , Department of CSE, Er.Perumal Manimekalai College of Engineering, Hosur

Abstract: ***The Internet of Vehicles (IoV), encompassing connected and autonomous vehicles, enables seamless communication among vehicles, infrastructures, and smart devices via vehicle-to-everything (V2X) technologies. While these advancements improve transportation efficiency and safety, they also expand the attack surface, increasing vulnerability to various cyber threats targeting both intra-vehicle and external networks. To address these security challenges, we propose a novel tree-based multi-tiered ensemble Intrusion Detection System (IDS) that integrates signature-based and anomaly-based detection methods. The system leverages advanced machine learning models—XGBoost, LightGBM, CatBoost, and Random Forest—organized in an ensemble framework called Leader Class and Confidence Decision Ensemble (LCCDE). This framework selects the best-performing model for each attack type and utilizes prediction confidence for accurate detection. Experimental evaluations on CAN-intrusion, Car-Hacking, and CICIDS2017 datasets demonstrate the system's high detection accuracy, effective zero-day attack recognition, and real-time feasibility with a low processing overhead. These results highlight the system's robustness in detecting a wide range of known and unknown cyber- attacks in IoV environments.***

Index Terms— Intrusion Detection System, Internet of Vehicles, CAN Bus, Ensemble Learning, Machine Learning, Anomaly Detection, Zero-Day Attacks, Cybersecurity.

I. INTRODUCTION

With the rapid advancement of Internet of Things (IoT) and Internet of Vehicles (IoV) technologies, modern vehicles— such as Autonomous Vehicles (AVs) and Connected Vehicles (CVs)—are becoming highly interconnected systems that rely heavily on both internal and external networks. IoV typically comprises intra-vehicle networks (IVNs), like the Controller Area Network (CAN) bus, which enables communication among Electronic Control Units (ECUs), and external networks powered by Vehicle-to- Everything (V2X) communication technologies. V2X includes Vehicle-to-Vehicle (V2V), Vehicle-to- Infrastructure (V2I), and Vehicle-to-Network (V2N) communications, facilitating interactions between vehicles, roadside units, infrastructure, and smart devices.

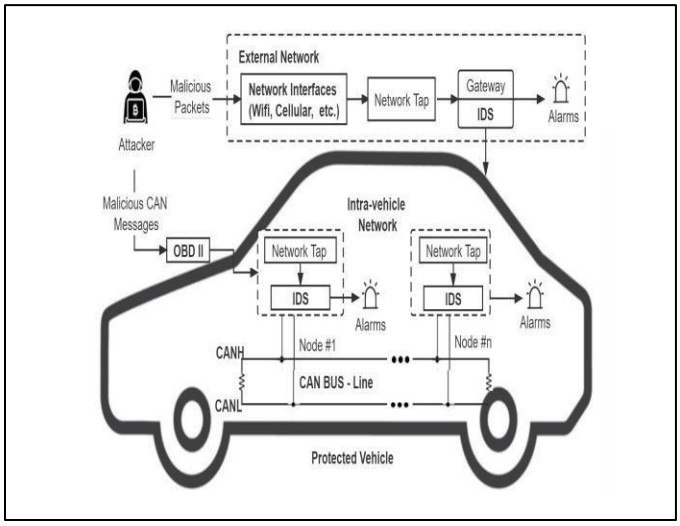


Fig. 1. The IDS-protected vehicle architecture(Decision-Based Framework).

Figure 1 depicts the IoV attack scenarios, including IVN and external network attacks. Intrusion Detection Systems (IDSs) have been developed as promising solutions for de tecting intrusions and defending Internet of Vehicles (IoV) systems and smart automobiles from cyber-attacks .However, this growing connectivity significantly increases the attack surface, making modern vehicles susceptible to a wide range of cyber threats. The CAN bus, due to its lack of encryption and authentication mechanisms, is particularly vulnerable to internal attacks such as Denial of Service (DoS), spoofing, and fuzzy attacks. Similarly, external vehicular networks are exposed to conventional cyber-attacks like GPS spoofing, sniffing, brute-force, SQL injection, and cross-site scripting (XSS), which can compromise vehicle operations and endanger human safety.Given the limitations of traditional security mechanisms in the vehicular context—such as high latency or incompatibility with CAN's real-time requirements—Intrusion Detection Systems (IDSs) have emerged as a promising defense strategy. IDSs monitor network traffic to detect malicious activities in both IVNs and external networks. Recent research has increasingly focused on enhancing IDSs using Machine Learning (ML) techniques, which can classify and detect various attack patterns with high accuracy. Several advanced IDS frameworks have been proposed to address the diverse and complex nature of cyber threats in IoV environments. These include ensemble models combining multiple ML algorithms, hybrid architectures for detecting both known and zero-day attacks, and optimized IDSs using techniques like feature selection and Bayesian tuning. Datasets such as CAN-intrusion and CICIDS2017 are widely used to train and validate these systems, providing real-world data from both internal and external vehicular communications.

As vehicles continue to evolve into intelligent and autonomous systems, the dependency on digital communication and cloud-based infrastructure has grown exponentially. This integration brings about tremendous benefits such as enhanced traffic management, improved passenger safety, real-time navigation, predictive maintenance, and infotainment. However, it also introduces a complex cybersecurity landscape that traditional automotive security mechanisms are ill-equipped to handle.The open and dynamic nature of IoV environments allows adversaries to exploit both physical and remote access points, targeting vulnerabilities within sensors, ECUs, cloud platforms, or even mobile applications associated with the vehicle. The increasing reliance on Over-the-Air (OTA) updates and third- party services further elevates the risk of infiltration, malware injection, and data theft.

Moreover, the heterogeneity of data in IoV—ranging from high-frequency sensor data to slower, structured network traffic—poses significant challenges in real-time threat detection and response. In this context, Machine Learning and Deep Learning techniques offer adaptive and scalable solutions capable of analyzing large volumes of data, recognizing subtle anomalies, and detecting both known and previously unseen attack patterns.To strengthen vehicular cybersecurity, there is a growing need for robust, lightweight, and real-time IDSs that can operate efficiently within resource-constrained environments like ECUs. Researchers are exploring multi-layer IDS architectures that combine rule-based detection with behavioral analysis, integrating cloud and edge computing paradigms for distributed detection and faster response times.
In summary, safeguarding IoV systems demands a holistic approach that encompasses secure communication protocols, intelligent IDS mechanisms, and constant updates to threat intelligence. As the automotive industry accelerates toward full autonomy and smart mobility, cybersecurity must be treated as a fundamental design principle rather than an afterthought.

II. RELATED WORK

Recent research on Intrusion Detection Systems (IDS) for Internet of Vehicles (IoV) highlights a growing focus on securing both intra-vehicle (CAN-based) and external vehicular networks. Several studies have proposed machine learning (ML) and deep learning (DL) based IDS models to detect CAN intrusions. Techniques like SVM, KNN, SSAE, LSTM, and CNN have shown high detection accuracy but often suffer from high computational costs or limited real- time applicability. Notably, models like SAIDuCANT and OTIDS have been evaluated in real-time environments, demonstrating better feasibility.

On the external network front, approaches such as MLP, DBN, and decision tree-based IDSs have utilized benchmark datasets like NSL-KDD, CICIDS2017, and UNSW-NB15. However, many studies target specific attacks (e.g., DoS, Botnets), limiting their scope. Only a few methods, such as those by Ashraf et al. and Alheeti et al., attempt to address both intra-vehicle and external threats or detect zero-day attacks.Moreover, essential processes like feature engineering and model optimization, crucial for enhancing IDS efficiency and detection performance, are often neglected. Only six studies included feature engineering and just one focused on model optimization.
To address these limitations, our proposed MTH-IDS is designed to detect both known and unknown attacks across intra-vehicle and external networks. It emphasizes real-time applicability by incorporating data sampling, feature engineering, and model optimization to enhance detection accuracy, reduce false alarms, and ensure low computational overhead—making it more suitable for real-world deployment in vehicular networks.

ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 10, Issue 6, June 2025

III. PROPOSED LCCDE FRAMEWORK

A. System Overview

The purpose of this work is to develop an ensemble IDS framework that can effectively detect various types of attacks on both IVN and external vehicular networks. Figure 2 demonstrates the overall framework of the proposed system, consisting of two phases: model training and model prediction. At the model training stage, three advanced ML algorithms, XGBoost , LightGBM , and CatBoost , are trained on the IoV traffic dataset to obtain the leader models for all classes/types of attacks. At the model prediction stage, the class leader models and their prediction confidences are used to accurately detect attacks. The algorithm details are provided in this section.

B. Base Machine Learning Models.

The proposed system employs ensemble learning using advanced Gradient Boosting Decision Tree (GBDT) models—XGBoost, LightGBM, and CatBoost—to enhance performance and accuracy.Decision Trees (DTs) form the foundation, making decisions via a tree-like structure based on the divide-and-conquer method.GBDT builds multiple DTs iteratively, combining their predictions to improve accuracy.

Advanced GBDT Models:

1. XGBoost:

Enhances GBDT with regularization and second-order Taylor approximation to reduce overfitting.Offers low computational complexity and supports parallel execution for faster training.

2. LightGBM

Efficiently handles large-scale, high-dimensional data. Uses Gradient-based One-Side Sampling (GOSS) and Exclusive

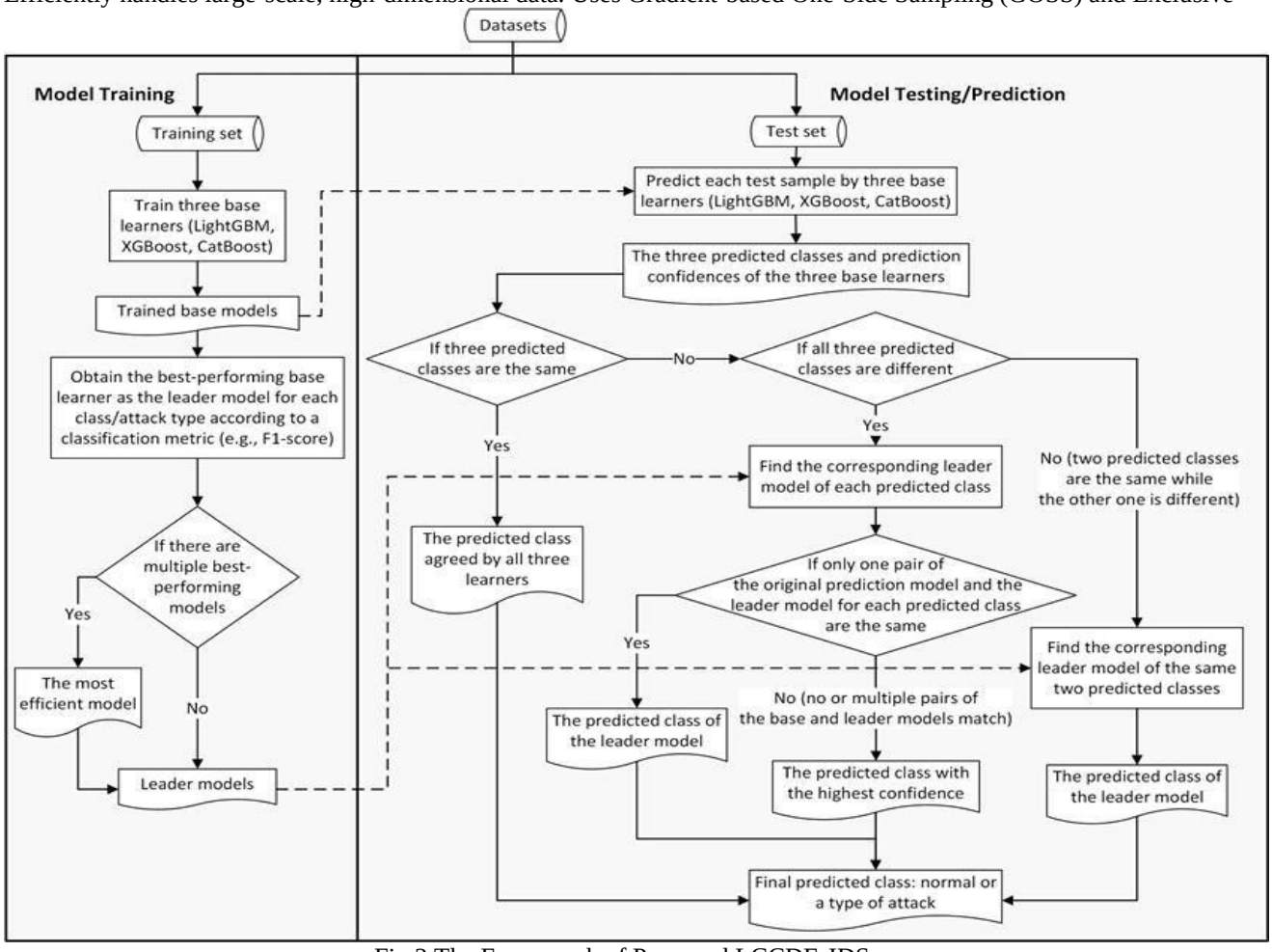


Fig.2.The Framework of Proposed LCCDE-IDS

3. CatBoost:
Designed to handle categorical features effectively.Employs symmetric trees, ordered boosting, and native support for all feature types to improve prediction speed, reduce overfitting, and simplify preprocessing.
Why These Models Were Chosen:
All three are robust, widely used ensemble models.They automatically perform feature selection, reducing preprocessing efforts.They are computationally efficient, support parallel and GPU execution.Their randomization in training allows for building diverse, generalizable ensemble models.

C. LCCDE: Proposed Ensemble Algorithm

The proposed LCCDE (Leader-Class Confidence-based Dynamic Ensemble) framework is an intelligent ensemble learning method developed to improve the accuracy and adaptability of machine learning models in detecting various types of cyber-attacks. The motivation behind this framework lies in the observation that different machine learning (ML) models perform variably when detecting different types of attacks—for instance, one model may be highly effective at identifying DoS attacks, while another may excel at detecting sniffing attacks. To overcome this inconsistency, LCCDE combines the strengths of three powerful gradient-boosting ML models—XGBoost, LightGBM, and CatBoost—to create an ensemble that dynamically selects the best model for each attack class.

Training Phase:

In the training phase, LCCDE operates in the following steps:

1. Training Base Learners: The three selected ML models (XGBoost, LightGBM, CatBoost) are independently trained on the training dataset.
2. Evaluating Performance: Each trained model is evaluated using F1-scores through cross-validation. The F1- score is chosen due to its effectiveness in handling class imbalances, which are common in network intrusion datasets.
3. Selecting Leader Models: For every class (normal or attack type), the model that achieves the highest F1-score is chosen as the leader model. In the case where multiple models have the same F1-score, the one with the fastest processing speed is selected.

Prediction Phase:

Once the leader models are determined, LCCDE proceeds with predicting the class of new, unseen data samples using these steps.

1. Initial Predictions: All three base learners generate predictions and associated confidence scores (probabilities) for each test sample.
2. Full Agreement Check: If all three models predict the same class, that prediction is accepted as the final result.
3. Full Disagreement Handling: If all three models predict different classes, the model that matches the leader model for its predicted class is trusted. If no match occurs, the prediction with the highest confidence score is chosen.
4. Partial Agreement Handling: If two models agree on a class and one differs, the leader model associated with the agreed-upon class is used to finalize the prediction.

these steps.1.Initial Predictions: All three base learners generate predictions and associated confidence scores (probabilities) for each test sample.2. Full Agreement Check: If all three models predict the same class, that prediction is accepted as the final result.
3. Full Disagreement Handling: If all three models predict different classes, the model that matches the leader model for its predicted class is trusted. If no match occurs, the prediction with the highest confidence score is chosen.
4. Partial Agreement Handling: If two models agree on a class and one differs, the leader model associated with the agreed-upon class is used to finalize the prediction.

Key Principles of LCCDE:

1. Utilization of Base Learners: It first generates predictions from all trained models.
2. Leader Model Reference: Uses pre-determined leader models to finalize decisions.
3. Confidence-Based Selection: If ambiguity arises, it relies on the model prediction with the highest confidence score.

Computational Complexity:

ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 10, Issue 6, June 2025

The computational complexity of the LCCDE framework is O(NCK), where:

Algorithm 1: Leader Class and Confidence Decision Ensemble (LCCDE) - Model Training

Input:
D_{train}: the training set,
$M = \{M_1, M_2, M_3\}$: the base ML model list, including M_1 = LightGBM, M_2 = XGBoost, M_3 = CatBoost,
$c = 1, 2, \ldots, n$: the class list for n different classes.
Output:
$M = \{M_1, M_2, M_3\}$: the trained base model list,
$LM = \{LM_1, LM_2, \ldots, LM_n\}$: the leader model list for all classes.

$M_1 \leftarrow Training(M_1, D_{train})$; // Train the LightGBM model
$M_2 \leftarrow Training(M_2, D_{train})$; // Train the XGBoost model
$M_3 \leftarrow Training(M_3, D_{train})$; // Train the CatBoost model
for $c = 1, 2, \ldots, n$ **do** // For each class (normal or a type of attack), find the leader model
  $Mlist_c \leftarrow BestPerforming(M_1, M_2, M_3, c)$; // Find the best-performing model for each class (*e.g.,* has the highest F1-score)
  if $Len(Mlist_c) == 1$ **then** // If only one model has the highest F1
    $LM_c \leftarrow Mlist_c[0]$; // Save this model as the leader model for the class c
  else // If multiple ML models have the same highest F1-score
    $LM_c \leftarrow MostEfficient(Mlist_c)$; // Save the fastest or most efficient model as the leader model for the class c
  end
  $LM \leftarrow LM \cup \{LM_c\}$; // Collect the leader model for each class
end

N = size of the dataset,
C = number of classes,
K = complexity of the base ML models.

Algorithm 2: Leader Class and Confidence Decision Ensemble (LCCDE) - Model Prediction

Input:
D_{test}: the test set,
$M = \{M_1, M_2, M_3\}$: the trained base ML model list, including M_1 = LightGBM, M_2 = XGBoost, M_3 = CatBoost,
$c = 1, 2, \ldots, n$: the class list for n different classes.
Output:
L_{test}: the prediction classes for all test samples in D_{test}.

for each data sample $x_i \in D_{test}$ **do** // For each test sample
  $L_{i1}, p_{i1} \leftarrow Prediction(M_1, x_i)$; // Use the trained LightGBM model to predict the sample, and save the predicted class & confidence
  $L_{i2}, p_{i2} \leftarrow Prediction(M_2, x_i)$; // Use XGBoost to predict
  $L_{i3}, p_{i3} \leftarrow Prediction(M_3, x_i)$; // Use CatBoost to predict
  if $L_{i1} == L_{i2} == L_{i3}$ **then** // If the predicted classes of all the three models are the same
    $L_i \leftarrow L_{i1}$; // Use this predicted class as the final predicted class
  else if $L_{i1} != L_{i2} != L_{i3}$ **then** // If the predicted classes of all the three models are different
    for $j = 1, 2, 3$ **do** // For each prediction model
      if $M_j == LM_{L_{i,j}}$ **then** // Check if the predicted class's original ML model is the same as its leader model
        $L\_list_i \leftarrow L\_list_i \cup \{L_{i,j}\}$; // Save the predicted class
        $p\_list_i \leftarrow p\_list_i \cup \{p_{i,j}\}$; // Save the confidence
      end
    end
    if $Len(L\_list_i) == 1$ **then** // If only one pair of the original model and the leader model for each predicted class is the same
      $L_j \leftarrow L\_list_i[0]$; // Use the predicted class of the leader model as the final prediction class
    else // If no pair or multiple pairs of the original prediction model and the leader model for each predicted class are the same
      if $Len(L\_list_i) == 0$ **then**
        $p\_list_i \leftarrow \{p_{i1}, p_{i2}, p_{i3}\}$; // Avoid empty probability list
      end
      $p\_max_i \leftarrow \max(p\_list_i)$; // Find the highest confidence
      if $p\_max_i == p_{i1}$ **then** // Use the predicted class with the highest confidence as the final prediction class
        $L_i \leftarrow L_{i1}$;
      else if $p\_max_i == p_{i2}$ **then**
        $L_i \leftarrow L_{i2}$;
      else
        $L_i \leftarrow L_{i3}$;
      end
    end
  else // If two predicted classes are the same and the other one is different
    $n \leftarrow mode(L_{i1}, L_{i2}, L_{i3})$; // Find the predicted class with the majority vote
    $L_i \leftarrow Prediction(M_n, x_i)$; // Use the predicted class of the leader model as the final prediction class
  end
  $L_{test} \leftarrow L_{test} \cup \{L_i\}$; // Save the predicted classes for all tested samples;
end

D. Performance Evaluation

To develop the proposed IDS, the models were implemented using Scikit-learn, Xgboost , Lightgbm , and Catboost libraries in Python. The proposed LCCDE framework is evaluated on two public benchmark IoV network security datasets, Car-Hacking and CICIDS2017 datasets, representing the IVN and external network data, respectively. The Car-Hacking dataset is created by transmitting CAN messages into a real vehicle's CAN bus. It has nine features (i.e., CAN ID and the eight bits of the CAN message data field) and four types of attacks (i.e., DoS, fuzzy, gear spoofing, and Revolutions Per Minute (RPM) spoofing attacks). The CICIDS2017 dataset is a state-of-the-art general cyber-security dataset including the most updated types of attacks (i.e., DoS, sniffing, brute-force, web-attacks, botnets, and infiltration attacks). To evaluate the proposed LCCDE model, five-fold cross validation is used in the training process to select leader class models, and 80%/20% hold-out validation is then used in the testing process to evaluate the model on the unseen test set. As network traffic data is often highly imbalanced and contains only a small proportion of attack samples, four performance measures, including accuracy, precision, recall, and F1-scores, are utilized to evaluate the model performance . The execution time, including the model training and test time, is used to evaluate the efficiency of the model.

TABLE I
PERFORMANCE EVALUATION OF MODELS ON CAR-HACKING DATASET

Method	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 (%)**	**Execution Time (s)**
KNN	97.4	96.3	98.2	93.4	195.6
SVM	**96.5**	95.7	98.3	93.3	1345.3
LSMTM-AE	99.0	99.0	99.9	99.0	-
DCNN	99.93	99.84	99.84	99.81	-
LightBoost	99.9997	99.9997	99.9997	99.9997	10.7
XGBoost	99.9994	99.9994	99.9994	99.9994	45.3
CatBoost	99.9994	99.9994	99.9994	99.9994	88.6
Proposed LCCDE	**99.9997**	**99.9997**	**99.9997**	**99.9997**	**185.1**

TABLE II
PERFORMANCE EVALUATION OF MODELS ON CICIDS2017

Method	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 (%)**	**Execution Time (s)**
KNN	96.3	96.2	93.7	96.3	1558.3

SVM	98.82	98.8	99.95	98.8	135.1
LSMTM-AE	98.95	95.82	95.81	95.81	-
DCNN	99.80	99.75	99.89	99.70	278.6
LightBoost	99.794	99.795	99.794	99.792	14.3

IV. PROPOSED MTH-IDS FRAMEWORK

Four tree-based supervised learners — decision tree (DT), random forest (RF), extra trees (ET), and extreme gradient boosting (XGBoost) — used as multi-class classifiers for known attack detection;Therefore, tiers 1 and 3 of the MTH- IDS are designed for basic known and unknown attack detection functionalities, respectively, while tiers 2 and 4 are designed to optimize the base learners in tiers 1 and 3 for model performance enhancement. A comprehensive and robust IDS with both known and unknown attack detection functionalities can be obtained after the model learning and optimization procedures. Additionally, the quality of the used datasets can be improved by data pre-processing and feature engineering procedures to achieve more accurate attack detection.

A. CAN Intrusion Detection

The research on IDS development for IoV and connected vehicles has been considered critical in recent years. Many research works have a focus on detecting attacks on CAN based intra-vehicle networks. Alshammari et al. proposed an intrusion classification model to identify CAN intrusions on in-vehicle networks utilizing support vector machine (SVM) and k-nearest neighbors (KNN) algorithms. Barletta et al. proposed a distance-based IDS for CAN intrusion detection using a X–Y fused Kohonen network with the k-means algo rithm (XYF-K). The proposed method shows high accuracy on the CAN-intrusion dataset, but its main limitation is the high computational complexity. Olufowobi et al. proposed a specification-based real-time IDS named SAIDuCANT to detect the in-vehicle network attacks. The effectiveness of SAIDuCANT is evaluated on a synthetic dataset and the CAN intrusion dataset. Olufowobi et al. proposed an anomaly based IDS for CAN attack detection using the adaptive cumu lative sum (CUSUM) algorithm. This technique can effectively detect intrusions with low delay based on statistical changes. Lee et al.proposed an Offset Ratio and Time Interval based IDS (OTIDS) to detect CAN attacks in in-vehicle networks. They also created a CAN dataset by simulating DoS, fuzzy, and impersonation attacks for IDS evaluation. Deep learning (DL) methods are also widely used for intra-vehicle network IDS development. Lokman et al. proposed an unsupervised DL-based anomaly detection model named stacked sparse autoencoders (SSAEs) to discover anomalies in CAN-bus data for intra-vehicle network security enhancement. Song et al. proposed a deep convolutional neural network (DCNN) method named Reduced Inception ResNet to detect intra- vehicle attacks and achieve high detec tion performance on the CAN-intrusion dataset. Ashraf et al. proposed a DL-based IDS for IoV using a long-short term memory (LSTM) autoencoder algorithm. The effectiveness of the proposed model is evaluated on the CAN-intrusion-dataset and UNSW-NB15 dataset, representing in-vehicle network and external network datasets, respectively. DL methods can often achieve high accuracy, but they are computationally expensive due to high model complexity. Intrusion detection in IoV or external vehicular networks has also attracted significant attention. Alheeti et al.proposed an intelligent IDS using back-propagation neural networks to detect DoS attacks in external vehicular networks using the Kyoto 2006+ dataset, but did not consider other attack types. Rosay et al. proposed a multi- layer perceptron (MLP) based network IDS for cyber-attack detection in IoT and connected vehicles. The proposed model has been imple mented on an automotive microprocessor, and its performance is evaluated on the two variants of the CICIDS2017 dataset. Aswal et al. analyzed the applicability of six classical ML algorithms for Bot attack detection on IoV. They used the Bot attack files in the CICIDS2017 dataset to represent the Botnets in vehicular networks, but they did not consider 3 other attacks.

C.VEHICULAR NETWORKS, VULNERABILITIES, AND IDS DEPLOYMENT

i. Vulnerabilities of Intra-vehicle Networks

Modern vehicles often contain 70-100 ECUs that are in vehicle components used to enable various functionalities . CAN is a bus communication protocol that defines an international standard for efficient and reliable intra-vehicle communications

among ECUs. A CAN-bus is built based on differential signaling and comprises a pair of channels, CAN High and CAN-Low, representing the two signals, 1 and 0, respectively .CAN is the most common type of IVN due to its low cost and complexity, high reliability, noise resistance, and fault-tolerance properties. However, CANis vulnerable to various cyber threats due to its broadcast transmission strategy, lack of authentication and encryption, and unsecured priority scheme.CAN messages, or packets, are transmitted via CAN-bus. The data frame is the most important type of CAN packet used to transmit user data . the structure of a CAN packet, which consists of seven fields : start of frame, arbitration field, control field, data field, CRC (cyclic redundancy code) field, acknowledge (ACK) field, and end of frame. Among all fields, the data field with the size of 0-8 bytes is the most important and vulnerable one, since it contains the actual transmitted data that determines the node actions .An attacker can intrude or take control of a vehicle by injecting malicious messages into the data field of CAN packets, resulting in compromised nodes or vehicles; so-called message injection attacks. Message injection attacks are the primary type of intra vehicle attack and can be further classified as DoS attacks, fuzzy attacks, and spoofing attacks by their objectives . In DoS attacks, a CAN is flooded with massive high-priority mes sages to cause latencies or unavailability of other legitimate messages.

ii. Vulnerabilities of External Vehicular Networks

In a similar fashion, V2X technology enables interactions and communications between vehicles and other IoV entities, including pedestrians, infrastructures, smart devices, and net work systems .With the increasing connectivity of modern IoV, external vehicular networks are becoming large networks that involve various other networks and devices. Thus, external vehicular networks are vulnerable to various general cyber threats because each vehicle or device is a potential entry point for intrusions. Typical attacks in IoV include DoS, GPS spoofing, jamming, sniffing, brute-force, Botnets, infiltration, and web attacks. The description and IoV scenarios of these common external vehicular network attacks are summarized in Table III.

TABLE III
COMMON ATTACK TYPES ON EXTERNAL VEHICULAR NETWORKS

Attack Type	**Description and IoV Scenarios**
DOS	Send a large number of requests to exhaust the compromised nodes' resources, causing vehicle unavailability or accidents.
GPS Spoofing	Masquerade as authorized IoV users to provide a node with false information, like false geographic information, therefore causing fake evidence, event delay, or property losses.
Jamming	Jam signals to prevent legitimate IoV devices from communicating with connected vehicles.
Sniffing	Capture vehicular network packets to steal confidential or sensitive information of vehicles, users, or enterprises.
Brute-force	Crack passwords in vehicle systems to take control of vehicles or machines and perform malicious actions.
Botnets	Infect multiple connected vehicles and IoV devices with Bot viruses to breach them and launch other attacks.
Infiltration	Traverse the compromised vehicle systems and create a backdoor for future attacks.
Web Attack	Hack IoV servers or web interfaces of connected vehicles to gain confidential information or perform malicious actions.

D.System Architecture

The purpose of this work is to develop an IDS that can protect both intra-vehicle and external networks from being breached by the various common attacks. In this paper, a novel multi-tiered hybrid IDS is proposed to detect both known and unknown cyber attacks on vehicular networks with optimal performance. Fig. 3 demonstrates the architecture of the proposed system,

ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 10, Issue 6, June 2025

comprising four main stages: data pre-processing, feature engineering, a signature- based IDS, and an anomaly-based IDS. Firstly, intra-vehicle and external network traffic datasets are collected for the purpose of system performance evaluation on both types of vehicular networks. Data pre-processing consists of a k- means-based cluster sampling method used to generate a highly-representative subset, and a SMOTE method used to avoid class-imbalance. In the feature engineering process, the datasets are processed by information gain-based and correlation-based feature selection methods to remove

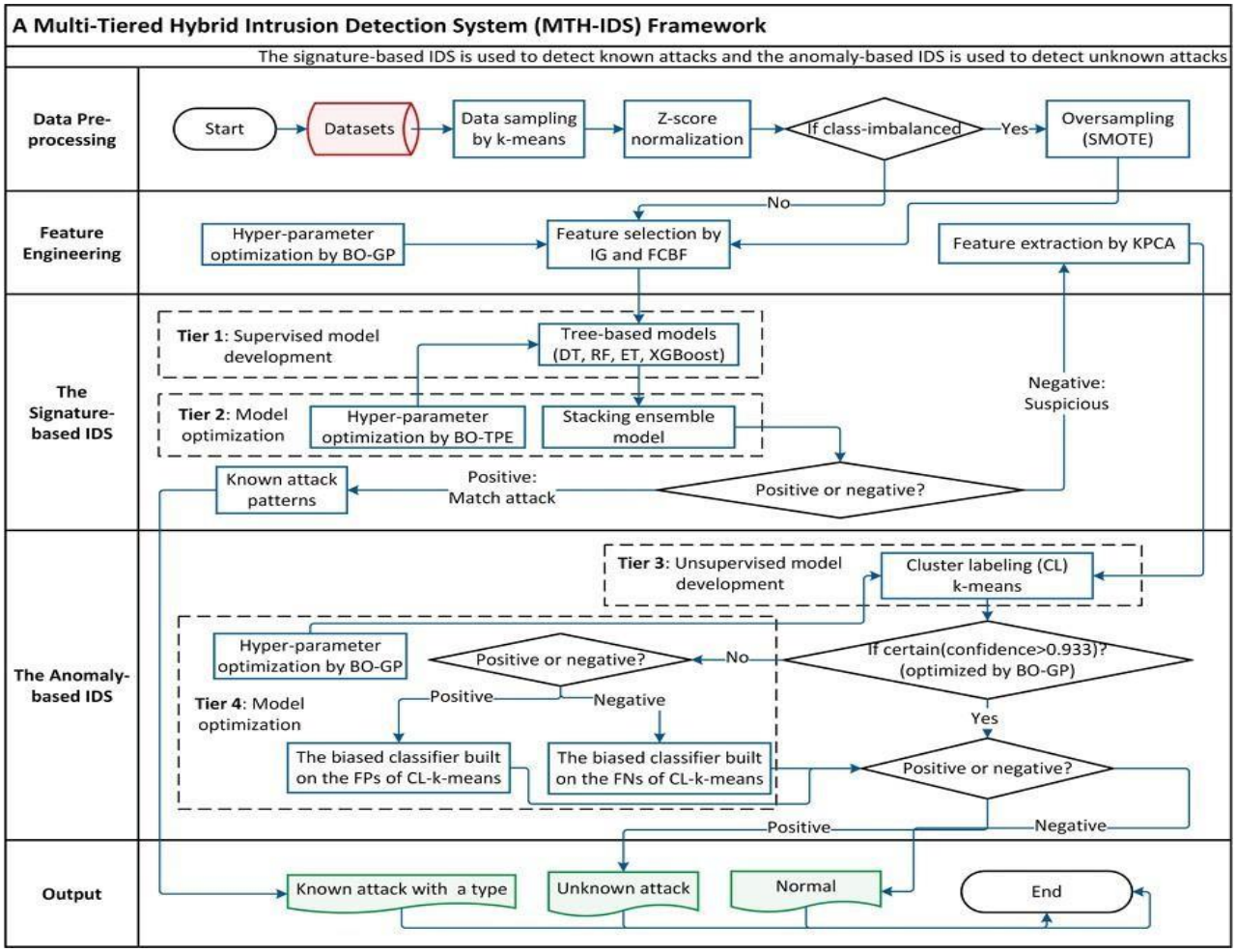


Fig. 3. The framework of the proposed MTH-IDS.

irrelevant and redundant features, and then passed to the KPCA model to further reduce dimensionality and noisy features. The proposed data pre-processing and feature engineering procedures can greatly improve the quality of the network data for more accurate model learning. The signature based IDS is then developed to detect known attacks by training four tree- based machine learners as the first tier of the proposed MTH- IDS: DT, RF, ET, XGBoost. In the second tier, a stacking ensemble model and the BO-TPE method are used to further improve the intrusion detection accuracy by combining the output of the four base learners from the first tier and optimizing the learners. In the next stage, an anomaly based IDS is constructed to detect unknown attacks. In the anomaly-based IDS, the suspicious instances are passed to a cluster-labeling (CL) k-means model as the third tier to effectively separate attack samples from normal samples. The fourth tier of the MTH-IDS comprises the BO-GP method and two biased classifiers used to optimize the model and reduce the classification errors of the CL-k-means. Ultimately, the detection result of each test sample is returned, which could be a known attack with its type, an unknown attack, or a normal packet. To summarize the rationale behind the algorithms used in the proposed IDS, the brief description and performance impact of each algorithm are presented in Table IV.

ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 10, Issue 6, June 2025

E. Performance Evaluation

For the purpose of intra-vehicle network IDS development, the first used dataset is the CAN-intrusion- dataset proposed in 2018 . The dataset is generated by logging CAN traffic via the OBD-II port of a vehicle when CAN attacks are launched. The features of this dataset include timestamp, CAN ID, data length code (DLC), and the 8-bit data field of CAN packets . Since the feature "timestamp" has a strong correlation with cyber-attack simulation periods and can lead to biased models and results, this feature was removed from the feature space. As the authors in created the CAN-intrusion-dataset by injecting attacks into random CANIDs, this CAN ID feature is retained. Based on the CAN

Stage	Algorithm	Rationale and Description	Performance Impact
Data pre-processing	K-means cluster sampling	Network traffic data is often large, while IoV devices often have limited computational power and resources. The k-means sampling method can generate highly representative subsets for more efficient training because the removed data is mostly redundant data.	Improve model training efficiency.
	SMOTE	Network traffic data is often imbalanced data because most data samples are collected under normal conditions in real-world vehicle systems. SMOTE can create high-quality samples for minority classes to avoid class-imbalance and ineffective classifiers.	Improve detection rate.
	Z-score	Different features often have different ranges, which can bias the model training. The Z-score method can normalize features to a similar scale and handle outliers.	Improve model accuracy and training efficiency.
Feature engineering	IG	For certain tasks like intrusion detection, many collected features can be irrelevant, causing additional training time. The IG method can remove those unimportant features.	Improve model training efficiency.
	FCBF	Certain features are redundant because they contain very similar information. FCBF can remove redundant features by calculating the correlation between each pair of features.	Improve model accuracy and training efficiency.
	KPCA	The anomaly-based IDSs are sensitive to the quality of features. KPCA can further extract the most relevant features to reduce dimensionality and noisy information.	Improve model accuracy and training efficiency.
The signature-based IDS	DT, RF, ET, and XGBoost	Tree-based ML algorithms often perform better than other ML algorithms on complex tabular data to which IoV data belong. Four tree-based supervised algorithms are used to train base classifiers for known intrusion detection.	Detect various types of known attacks.
	BO-TPE	The default hyper-parameters of ML algorithms often cannot return the best model. BO-TPE can optimize the models' hyper-parameters to obtain the optimized base classifiers.	Improve accuracy of known attack detection.
	Stacking	Ensemble models can often achieve higher accuracy than any single model. Stacking ensemble can combine the base classifiers to obtain a meta-learner with better performance.	Improve accuracy of known attack detection.
The anomaly-based IDS	CL-k-means	For unknown attack detection, CL-k-means can generate a sufficient number of normal and attack clusters to identify zero-day attacks from the newly arriving data.	Detect unknown attacks.
	BO-GP	CL-k-means has an important hyper-parameter, the number of clusters, k. BO-GP is an effective HPO method to optimize k and obtain the optimized CL-k-means model.	Improve accuracy of unknown attack detection.
	Two biased classifiers	CL-k-means may return many errors when detecting complex unknown attacks. Two biased classifiers are trained on the FPs and FNs of CL-k-means to reduce the errors.	Improve accuracy of unknown attack detection.

TABLE IV RATIONALE AND PERFORMANCE IMPACT OF EACH COMPONENT OF THE MTH-IDS

packet structure ,ten features extracted from the identifier field and data field of CAN packets, including CAN ID, DLC, were preliminarily selected for IDS development. To enhance IoV security, this work proposed a multi-tiered hybrid intrusion detection system (MTH-IDS) model that can detect various types of known and zero-day cyber-attacks on both intra-vehicle and external-vehicular networks for modern vehicles. The proposed MTH-IDS consists of two traditional ML stages (data pre-processing and feature engineering) and four main tiers of learners utilizing multiple machine learning algorithms. Through data pre-processing and feature engi neering, the quality of the input data can be significantly improved for more accurate model learning. The first tier of the proposed system consists of four tree-based supervised learners used for known attack detection, while the second tier comprises the BO-TPE and stacking models for supervised base learner optimization to achieve higher accuracy. The third tier consists of a novel CL-k-means unsupervised model used for unknown/zero-day attack detection. Lastly, BO-GP and two biased classifiers are used to construct the fourth tier for unsupervised learner optimization. The four tiers of learning models enable the proposed MTH-IDS to achieve optimal performance for both known and unknown attack detection in vehicular networks. Through the performance evaluation of the proposed IDS on the two public datasets that represent intra-vehicle and external vehicular network data, the proposed system can effectively detect various types of known attacks with accu racies of 99.99% and 99.88% on the CAN-intrusion-dataset and CICIDS2017 dataset, respectively. Moreover, the

proposed system can detect various types of unknown attacks with average F1-scores of 0.963 and 0.800 on the CAN-intrusion dataset and CICIDS2017 dataset, respectively. The experimen tal results on a vehicle-level machine also show the feasibility of the proposed system in real-time environments. In future work, the proposed anomaly-based IDS framework can be further improved by doing research on other unsupervised learning and online learning methods.

V. PROPOSED TREE-BASED INTELLIGENT IDS FRAMEWORK

A. Problem statement

Due to the fact that the AV systems are vulnerable to many types of network threats through different communication mediums, a comprehensive IDS system is proposed to be im plemented for both intra-vehicle and external communication networks. This is done to better protect not only the vehicle components but also all the IoT devices involved in the entire IoV.

The proposed IDS should be able to detect various common intrusions launched on CAN bus and external networks. The main attacks to be considered in this work include DoS attacks on both the intra-vehicle and external communication networks, fuzzy attacks and spoofing attacks on the CAN- bus, sniffing, brute force and web attacks launched on the external networks.

The IDS should have a high detection rate to effectively identify most of the attacks and low computational time to improve efficiency.

B. IDS system overview and architecture

To provide protection for both the intra-vehicle and external communications, the proposed IDS is implemented in multiple locations within the AV system. To detect threats on the CAN bus and secure it, the IDS can be placed on the top of the CAN bus to process every transmitted message and ensure the nodes are not compromised. Alternatively, the proposed IDS can be placed inside the gateway to secure the

Fig.4.The proposed IDS-protected AV architecture

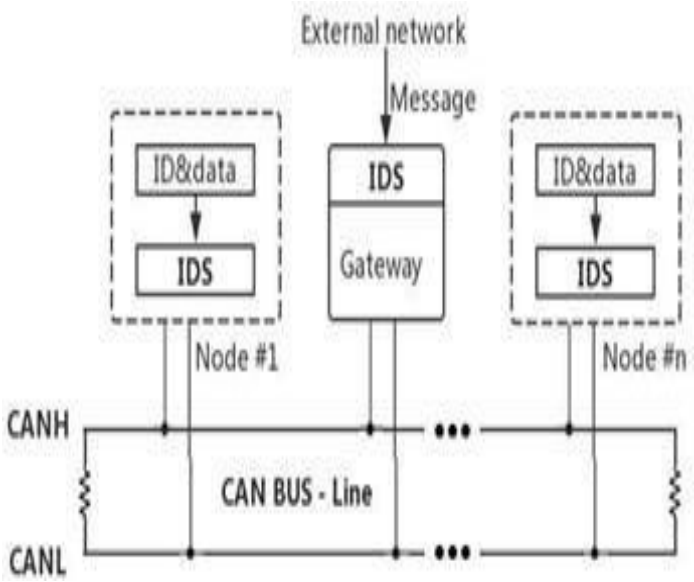
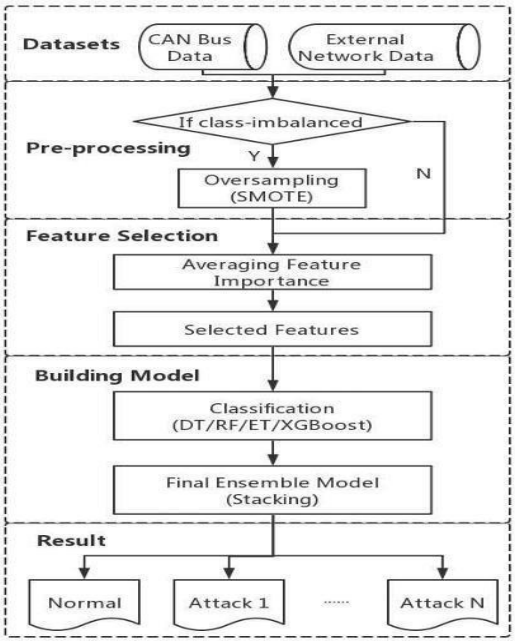


Fig.5. The proposed IDS Framework

ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 10, Issue 6, June 2025



external communication networks.The topology of IDS implementation on vehicle systems is shown in Fig. 1. On the CAN bus, all the nodes are linked, and when a node receives a message mainly consisting of an ID and data field sent by another device connected to the CAN bus, the message is passed through the IDS to check when the signal line of the CAN bus changes from CANH to CANL. Similarly, when a message is transmitted from external network to intranet, it is passed through the IDS inside the gateway and checked. To design the IDS, Fig. 5 provides an overview of the tree based ML classification framework for intrusion detection. The process of the proposed model is as follows. Firstly, suffi cient network traffic data is collected. Secondly, if the classes of the data set are imbalanced, oversampling is implemented to reduce its impact. At the next stage, feature selection based on averaging feature importance is done to reduce computational cost. After that, four base-models are built to be the input of the stacking ensemble model. At the end, the final model is built to classify the data.

VI. CONCLUSION

In this paper, the growing security concerns in Internet of Vehicles (IoV) networks were addressed through the integration of machine learning techniques in intrusion detection systems (IDS). With the rapid evolution of cyber threats targeting both intra-vehicle (CAN bus) and external vehicular networks, it is critical to adopt intelligent and adaptive security solutions. The proposed IDS frameworks— LCCDE and MTH-IDS—demonstrate innovative approaches that blend ensemble learning, multi-tiered model architectures, and hybrid methodologies to enhance detection performance.

The LCCDE model introduces a leader-class-based strategy, selecting the most effective ML models for each attack type, while incorporating prediction confidence to refine final classification decisions. This ensemble strategy, using XGBoost, LightGBM, and CatBoost, achieves near- perfect F1-scores on benchmark datasets, highlighting its robustness and adaptability.Similarly, the MTH-IDS framework incorporates a structured four-tier learning model combining supervised, unsupervised, and optimization techniques to handle both known and zero-day attacks.

The incorporation of BO-TPE, BO-GP, and clustering- based models strengthens detection accuracy and generalization across dynamic vehicular environments. Furthermore, the integration of data preprocessing techniques such as SMOTE and feature selection not only mitigates class imbalance but also reduces computational overhead, making the system suitable for real-time deployment.

Performance evaluations conducted on the CAN-Intrusion and CICIDS2017 datasets validate the effectiveness of the proposed systems, with accuracy rates approaching 100% and substantial reductions in processing time. Compared to existing literature, the proposed approaches consistently outperform in accuracy, detection rate, and false alarm reduction.

In future work, the effectiveness of these IDS frameworks can be further improved by exploring advanced unsupervised and online learning methods to adapt to new threat patterns in real-time. Additionally, employing optimization techniques such as Particle Swarm Optimization (PSO) and Bayesian Optimization for hyperparameter tuning could significantly enhance performance and scalability across different vehicular platforms.

REFERENCES

[1] H. Bangui and B. Buhnova, "Recent Advances in Machine-Learning Driven Intrusion Detection in Transportation: Survey," Procedia Com put. Sci., vol. 184,
pp. 877–886, 2021.
[2] O. Y. Al-Jarrah, C. Maple, M. Dianati, D. Oxtoby, and A. Mouzakitis, "Intrusion Detection Systems for Intra-Vehicle Networks: A Review," IEEE Access, vol. 7, pp. 21266– 21289, 2019.
[3] L. Yang and A. Shami, "A Transfer Learning and Optimized CNN Based Intrusion Detection System for Internet of Vehicles," in 2022 IEEE Int. Conf. Commun. (ICC), 2022, pp. 1–6.
[4] K. Kim, J. S. Kim, S. Jeong, J.-H. Park, and H. K. Kim, "Cybersecurity for autonomous vehicles: Review of attacks and defense," Comput. Secur., vol. 103, p. 102150, 2021.
[5] L. Yang, A. Moubayed, and A. Shami, "MTH-IDS: A Multitiered Hybrid Intrusion Detection System for Internet of Vehicles," IEEE Internet Things J., vol. 9, no. 1, pp. 616–632, 2022.
[6] J. Jiang, F. Liu, W. W. Y. Ng, Q. Tang, W. Wang, and Q.-
V. Pham, "Dynamic Incremental Ensemble Fuzzy Classifier for Data Streams in Green Internet of Things," IEEE Trans. Green Commun. Netw., pp. 1-14, 2022.
[7] M. Injadat, A. Moubayed, A. B. Nassif, and A. Shami, "Machine learning towards intelligent systems: applications, challenges, and op portunities," Artif. Intell. Rev., 2021.
[8] L. Yang, D. M. Manias, and A. Shami, "PWPAE: An Ensemble Framework for Concept Drift Adaptation in IoT Data Streams," in proc. 2021 IEEE Glob. Commun. Conf.,
pp. 1–6, 2021.
[9] L. Yang , A. Moubayed, A. Shami, P. Heidari, A. Boukhtouta, A. Larabi, R. Brunner, S. Preda, and D. Migault, "Multi-Perspective Content Delivery Networks Security Framework Using Optimized Unsupervised Anomaly Detection," IEEE Trans. Netw. Serv. Manag., vol. 19, no. 1,
pp. 686-705, 2022.
[10] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.
[11] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," Adv. Neural Inf. Process. Syst., vol. 2017-December, no. Nips, pp. 3147–3155, 2017.
[12] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: Unbiased boosting with categorical features," Adv. Neural Inf. Process. Syst., vol. 2018-December, no. Section 4, pp. 6638–6648, 2018.
[13] E. Seo, H. M. Song, and H. K. Kim, "GIDS: GAN based Intrusion Detection System for In-Vehicle Network," 2018 16th Annu. Conf. Privacy, Secur. Trust, pp. 1–6, 2018.
[14] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in Proc. Int. Conf. Inf. Syst. Secur. Privacy, 2018, pp. 108–116.
[15] M. Injadat, A. Moubayed, A. B. Nassif, and A. Shami, "Multi-Stage Optimized Machine Learning Framework for Network Intrusion Detec tion," IEEE Trans. Netw. Serv. Manag., vol. 4537, no. c, pp. 1–14, 2020.
[16] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," Veh. Commun., vol. 21, p. 100198, 2020.
[17] R. Zhao et al., "A Novel Intrusion Detection Method Based on Lightweight Neural Network for Internet of Things," IEEE Internet Things J., vol. 9, no. 12, pp. 9960- 9972, 2022.
[18] L. Yang, A. Moubayed, I. Hamieh, and A. Shami, "Tree- Based Intelli gent Intrusion Detection System in Internet of Vehicles," in proc. 2019 IEEE Glob. Commun. Conf., pp. 1– 6, 2019.
[19] W. Elmasry, A. Akbulut, and A. H. Zaim, "Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic," Comput. Networks, vol. 168, 2020.
[20] Z. Chen, M. Simsek, B. Kantarci, and P. Djukic, "All Predict Wisest Decides: A Novel Ensemble Method to Detect Intrusive Traffic in IoT Networks," in proc. 2021 IEEE Glob. Commun. Conf., pp. 1–6, 2021.
[21] L. Yang and A. Shami, "A Lightweight Concept Drift Detection and Adaptation Framework for IoT Data Streams," IEEE Internet Things Mag., vol. 4, no. 2, pp. 96–101, 2021.

[22] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," Neurocomputing, vol. 415, pp. 295–316, 2020.
[23] A. Alshammari, M. A. Zohdy, D. Debnath, and G. Corser, "Classifica tion Approach for Intrusion Detection in Vehicle Systems," Wirel. Eng. Technol., vol. 09, no. 04, pp. 79–94, 2018.
[24] J. Ashraf, A. D. Bakhshi, N. Moustafa, H. Khurshid, A. Javed, and A. Beheshti, "Novel Deep Learning-Enabled" IEEE Trans. Intell. Transp. Syst., pp. 1–12, 2020.