Federated Learning Approaches for Collaborative Threat Detection in Autonomous Vehicle Networks

By Dr. Xiaojing Wang

Professor of Electrical and Computer Engineering, University of Illinois Urbana-Champaign (UIUC)

1. Introduction

The robustness of the AV should be guaranteed at multiple levels: (i) at the isolated level of a particular AV in response to cryptic communications from sensors and actuators and against stealthy attacks of methods; (ii) at the level of all distributed components (collaborative) of the AV fleet (including the remote operator or AV operational services) against threats that are capable of affecting the global functionality of the system of the AVs. It is clear that the security and safety of the AV fleet becomes crucial in the event of the verification of infrastructure-grade services (including human safety and private data security). The deployment and efficient functioning of the installations of the AV fleet are likely to occur through the establishment of future cooperative testbeds for safety assessment aimed at creating a completely acceptable technology by the corresponding standards (e.g., necessary conditions of the ISO 26262:2018 and the language of the SOTIF standard: preliminary work of the ISO/PAS 21448).

The evolution of information and communication technologies (ICT) has resulted in a significant increase in the deployment of cyber-physical systems (CPS) in various application domains, such as smart grids, infrastructure management systems, and many others. Among the most popular CPS are the so-called vehicles with various degrees of autonomy (or automated vehicles, AV). The latter include conventional vehicles with different advanced driver-assistance systems (ADAS), as well as fully autonomous vehicles (fAV). The proper operation of AV requires the coordination of various subsystems and microservices, such as localization, perception, and artificial intelligence (AI) for purposes of data fusion, prediction, planning, motion control, and external interfaces. Due to the variety and complexity of operational conditions, AV may be subjected to severe security and safety threats arising from different levels of adversarial intrusion (e.g., from jamming, spoofing, man-in-the-middle, false data injection, etc.) to the information methodologies and architectures, to the deviation from the expected vehicle operational profile from a software engineering perspective.

1.1. Background and Motivation

With recognition of these high-level security challenges, what approaches will effectively assist nations to develop the necessary capabilities that enable vehicle agencies to trust that their assets are not compromised? This chapter provides the framework for solutions that directly address system reliability and trust through enhanced federated learning and distributed sensing algorithms.

Potential adversaries could exploit system vulnerabilities to intentionally degrade performance, set suboptimal operating conditions, or intentionally disrupt critical infrastructure for political, economic, and social goals. In the context of security, considerations of defense, resilience, and safety were explicitly included as part of the intelligent connected vehicles (ICV) investment title found in the Fixing America's Surface Transportation Act, enacted into law in December 2015. That bill formally recognized the need to investigate and design novel strategies to robustly evolve vehicle systems that can build and operate in networked coalitions.

Connected, autonomous vehicles of the future are anticipated to evolve into intelligent and cooperative systems capable of understanding the physical and digital world around them. Achieving this vision demands a spectrum of capabilities including sophisticated sensors, distributed computing devices, robust digital communication, and vehicle fleets with shared information about their environment, state, and operating policies. Each of these capabilities and their hardware and software constituents can be susceptible to diverse risks and failures that destabilize performance at inopportune moments.

1.2. Research Objectives

Our goals are intertwined to address the challenges and threats restricted to the vehicle networks and the performance is measured with suitable multi-performance metrics. The human-in-the-loop core concept suggests that humans should be retained in the decision loop in some manner pertinent to the decision making context. Our validation of the localized threat detection analysis and the collision risk estimation methods involves the subject matter experts and other experts, ensuring that human safety concerns take precedence over automation with limited good behavior. Finally, we hope to unravel some of the mysteries surrounding "what is really the best approach to reaching optimal hierarchical decision making and information fusion?" This will contribute to answering questions such as the level of trust and confidence in machine decision making and automation.

The primary objectives in this research study are as follows: Scalable and efficient localized threat detection: Develop methodology and design threat detection approaches in the vehicle networks, such that the overhead on the computational nodes is controlled, and the localized environment perception and analysis for threat detection is achieved with high performance. Federated learning approaches: Converge the localized perception and analysis efforts in the vehicle network in a distributed manner using federated learning approaches. The need for centralized high-capacity processing, collection and sharing of training and testing data are avoided through this approach. The model weight exchange strategy is made robust to privacy preservation and cybersecurity threats in the vehicle networks. Reliability and robustness: To design localized environment perception and analysis using multi-sensory inputs that lead to robust recognition, robust diagnostic, error-free decision, and has mechanisms that prevent adversarial disruptions. Quantify collision risk: Quantify the collision risk based on the short and long term recognition and diagnostic uncertainties. Our research will uncover the complex relationships between the risk factors and discover the kinds of situations each of them is sensitized to.

1.3. Overview of Autonomous Vehicle Networks

The AVs are allowed to obtain indirect global threat state information while having no access to the local threat states of others. Such indirect global threat states will be requested and used in the training, and the AVs will eventually know the global state information from a detection strategy, thus avoiding privacy leaks.

AVs should make decisions based on local data they collect in real-time, and the collaborative system is acting upon the decisions made by all participants, thus making it a type of federated learning system. Four characteristics of federated learning advantageously adapt the constrained communication, shifting distribution, and verification of local data required by

AV networks, and to realize the collaborative threat detection and control in a practical manner.

Autonomous vehicles (AVs) have been an area of interest for years due to their convenience and attractiveness. The majority of existing research has explored driving safety, and the proposed technologies are currently being developed to turn the features provided by AVs into reality. Nonetheless, such potential features raise several nonfunctional requirements for the collaborative system in which AVs collaboratively detect and control potential threats, and demand concrete implementation realization.

2. Foundations of Federated Learning

Distributed Optimization. We consider the following problem in distributed optimization: minimize the function f(w) in such a way that f(w) is "close" to the minimum of some "target" function g(w*), where w are the parameters, w* is the optimum, and the functions f and g are IP twice continuously differentiable. To minimize f(w), we can utilize either first-order optimization and approximation of g(w*) or second-order optimization. The updates in distributed optimization at each machine t have the form wt+1 = wt - ηtst, where st is the search direction and ηt is the step size or learning rate. The search direction can be deterministic or randomly generated from an NHIP distribution. For example, in Distributed Stochastic Average Gradient (DSAG), st includes the gradient at the current time t and the algebraic average of the gradients in the ensemble at the last iteration considered.

We now provide foundational knowledge required to understand Federated Learning (FL). In particular, we discuss its relationship with Distributed Machine Learning (DML). This overview sets the stage for the subsequent sections.

2.1. Machine Learning Basics

where D^{train} contains m input-label pairs. A fundamental tradeoff in ML is to ensure h ≈ f and reducing the complexity of model H to prevent overfitting and maximizing generalization, which would prevent reduction in test error.

\[\hat z = \underset{z \epsilon Z}{\operatorname{argmin}} L(w; D^{train}) \]

Formally, for any defined cost function, L, and real learning task, with enough samples drawn from the underlying probability distribution, it is generally impossible to find the best hypothesis since we cannot draw all the samples of space Z. Hence, learning is approximated by minimizing the training error, empirical risk minimization (ERM), which defines the training error based on the available data.

where D contains input-label pairs, which is drawn from probability distribution P(X, Y), and Z contains the proposed output labels.

\[\hat z = \underset{z \epsilon Z}{\operatorname{argmin}} L(w; D) \]

Suppose we have a dataset, D, with input vectors, x, and corresponding labels, y. A learning algorithm typically tries to fit a mapping, h: x → y (h ≈ f). For a supervised algorithm, the mapping g which best matches the data can be thought of as a function in a hypothesis space, H, which maps the input vector to desired output. We often define some cost function, L: H → R, which measures how well the function in H agrees with the given dataset. We then search for some hypothesis, h ∈ H, that minimizes L.

2.2. Decentralized Learning

Furthermore, we will introduce Federated Averaging with Local Training (FALT) and Cooperative Learning with Delay (CLD) approaches dedicated to the dynamic, asynchronous, sample-starved scenarios in VCS. FALT equips with hyperparameters in federated learning and local training to coordinate between the optimization process under local data sparsity and slow alternations. Then, the optimization problem of seeking an agreement is formulated and reformulated. To solve such a non-convex, non-smooth, and large-scale optimization problem by the nature of 3D VCS, we make joint hierarchical feature learning and architectural parameter synchronization tasks be decomposed into several hierarchical coordination. These coordination tasks are reformulated by the combination of sync-delay bound constraint analysis and iterative solution strategies by which they could be transferred into several suboptimization problems with respect to each sample-pairing across different vehicles.

Decentralized learning, concurrently, is carried out in more than one local model and aims to find a consensus among the local models to achieve a global model. Unlike federated learning, decentralized learning can be carried out without the help of a central server. However, this is not the focus of our study. Our paper focuses on decentralized machine learning approaches in a partially connected network environment, where each vehicle learns from local data and models with some neighboring vehicles, and the neighboring vehicle set may temporally or spatially change. The training information asymmetry caused by different local data distributions among different vehicles and the communication delay due to the periphrastic updating mechanism could hinder their adaptation. In order to let vehicles with different local training data reach an agreement without sharing explicit information, we will make a decentralized simulation stack based on a fully connected neural network architecture to carry out decentralized hierarchical feature learning and architectural parameter synchronization based on a distributed concurrent simulation framework.

2.3. Privacy-Preserving Techniques

There are generally two classes of data privacy. The first class of privacy protection techniques leverages semi-trusted aggregation by using encryption schemes to at least prevent an entity from learning the contents of individual data until the completion of the aggregation. The second class of privacy-protection techniques are fully trusted techniques that leverage secure multi-party computation (MPC) to allow joint and distributed computations among the collaborating entities with the goal of keeping all the contents of their private data completely private from anyone without proper access. While the first class of techniques balances the trade-off between privacy and complexity much better, the second class of techniques presents a much better interpretation of privacy by treating data aggregation as private operations among all entities during the whole process. In this work, we consider MPC-based techniques for privacy-preserving multi-entity detection models.

Secure and privacy-preserving methods for multi-party data aggregation have been long studied in the context of data mining and statistics. In addition to providing security against adversarial models, such as semi-honest or malicious models, some approaches also consider privacy protection by not revealing any sensitive data of individual entities participating in data aggregation. In this study, we are particularly interested in privacy-preserving techniques that can be used in the context of collaborative and distributed threat detection in autonomous vehicle networks.

3. Threat Detection in Autonomous Vehicle Networks

A salient aspect of AV cybersecurity is the ability to detect and collaboratively address cyber threats likely to impact one or multiple AVs. Here, privacy-sensitive threat detection becomes crucial to protecting the safety integrity of AVs, while also mitigating the viability of stealthy abuse in practice. For instance, a vehicle can contain a stealthy attack node that can tamper with the data used for collaborative sensing of imminent traffic hazards or situations of co-threatening vehicles, causing one or both vehicles to collide with one another, thereby causing harm to their passengers or other bystanders nearby. Thus, due to vehicle deployments, the forms of data and operations conducted in edge computing, and the possible adversarial nodes and stealthy fault models, the threats connected to collaborative threat detection in the context of data and system security carry specific privacy implications. At the same time, these are the types of initiatives that the automotive industry is undertaking to ensure the safety and privacy preservation of smart transportation systems.

There are many sources of and reasons for concern about the security of autonomous vehicle systems. Both directly and indirectly, AVs handle sensitive data, including personally identifiable information (PII) about their passengers and others in the vicinity. To that end, cyber criminals could exploit and exfiltrate data associated with passengers and the vehicle platform itself to potentially attack not only individuals or organizations but also nearby vehicles and pedestrians. Additional vulnerabilities include the system interfaces connecting AVs to the broader transportation ecosystem, including cellular networks for infotainment, updates, vehicle-to-everything (V2X) coordination, and fleet management. These networks are exposed to evolving threat landscapes and can be attacked through vehicle stealthy misbehavior such as sniffing (e.g., eavesdropping for geolocational tracking), jamming (e.g., disabling sensors to induce disengagement or other unsafe actions), or replaying misleading data or injecting fabricated messages into the system.

3.1. Challenges and Requirements

Several paradigms have been proposed to execute machine learning processes in privacy-threatening distributed learning environments, including homomorphic encryption, secure multi-party computation, and federated learning. Among them, federated learning, an emerging technique in distributed learning, has recently gained considerable attention due to its ability to provide privacy-preserving and data-security benefits, since it uses local training and without the need for central data aggregation. In FL, each client (device) trains a local model with his private dataset and incorporates new updates in a global FL model, shared among all devices. This optimization process is done iteratively, with part of the training process being executed locally on each client to keep their data private. It is only necessary to communicate local model updates (or gradients) in each iteration, reducing the bandwidth and latency requirements between the clients and the coordinating server to a great extent.

Vehicles are becoming increasingly digitized and interconnected, relying on dozens of sensors and data sources for sensing, decision-making, and environmental perception. Cars produced now include advanced driver assistance systems (ADAS) that have cameras, radars, and ultrasonic sensors with connectivity to the internet and remote services (eCall). Moreover, the future of the automotive industry is promising, with a very noticeable increase in the development of technology for autonomous vehicles. This rapid growth is driven by data-driven development and the use of machine learning (especially deep learning) for perception capabilities in vehicles. This requires numerous high-quality labeled data for supervised learning to achieve high-quality models for tasks such as perception and control. Furthermore, cars are capable of data storage and have the ability to store diverse and detailed data, some of which are packaged in the form of data recordings, such as black boxes.

3.2. Existing Approaches

The approach we propose here will address the non-iid data at the Clients by using LOCO as a pre-selection step in Federated Learning. We call it "CL-Learning using LOCO for Federated Learning with Internet of Vehicles and Hyper-Edge Computing". In this iteration, B cells interact with the federated learning and local model training process of the neural network. Each cell in the federated process will carry out its local federated process. The system releases its specialization data for each cell participating in the model building of the federated learning process. The cloud will refer to it if the Client is needed again. The system network will build a new summary with new Client Releases, which contains only the most important changes made during the rounds.

The LOCO-FL approach will use LOCO as the main procedure along with Federated Learning for Multi-Party Data to address the problem of non-iid Clients in the federated learning of heavily distributed Clients. LOCO is used as a pre-selection procedure to extract a small representative set of clients, which is assumed to specialize in a certain portion of the knowledge. Each Client in the federated training process runs a local federated learning process trained on the LOCO model. A peer-client agreement occurs, with some local models summarized into a global model for consensus and synchronization of model parameters.

Several data sharing approaches have been proposed to improve model sharing across models already created by different model owners, while preserving privacy and security constraints. A simple solution is to share the model weights generated using Local Model Postprocessing. One method to ensure that published and shared models do not lead to overfitting of any portion of the data is to use differential privacy. Federated learning proposes an alternative to secure deep learning by sharing the knowledge used to train a model and removing the need to share the model itself.

4. Federated Learning for Threat Detection

The traditional centralized approach suffers from some critical issues due to the requirement on the centralized decision-making that all data should be aggregated and processed at the server side, causing significant communication overhead. The communication security is another critical concern in the centralized model. The already tightly-connected vehicle components, such as throttle, brake, and steering system, can be physically compromised through the connected vehicle system. To address the aforementioned issues, Distributed Threat Detection (DTD) systems have been designed at an early time. The DTD systems deploy physical sensors in each vehicle to carry out the intrusion detection. The detection performance and robustness can be enhanced, as the decision-making criteria are adaptively updated in response to the vehicle environment. However, the communication links among vehicles are not being fully exploited and the models would not serve as the infrastructure for other intelligent services in the V2V network. To address the issues currently available in traditional models, this paper examines the Federated Learning (FL) model for designing the collaborative threat detection model in the V2V network.

This section introduces a series of federated learning models adopting different network architectures for threat detection in the vehicle networks. We first provide background details about federated learning and the utilized network architectures. Next, two collaborative federated learning models for improving threat detection performance and robustness in dynamic V2V environment are presented. A hybrid federated learning system, FLEVINet, embedding a federated learning module into the centralized learning system, is then proposed to balance the identification performance in high and low SNR regions. The last model adopts a distributed federated learning approach in the V2V network. Centralized models contain a lot of false positive alerts and provide centralized decision capability based on the observed global patterns in traffic. Furthermore, with the collective intelligence and decision of all vehicles in the V2V network, the groundbreaking capabilities and services can be realized, which primarily depend on the real-time and secure access to contributed sensor data. However, currently there exist no standardized approaches for security monitoring and information sharing among vehicles, with the issues of intrusion into users' privacy and high communication overhead caused by a vast amount of transmitted data.

4.1. Concepts and Principles

The federated learning method is a method of distributed machine learning approach in which multiple participants (e.g., multiple connected vehicle networks) train central machine learning models in a coordinated way. Federated learning is designed for decentralized data networks where the data cannot be exchanged. It deploys models by executing several epochs of federated averaging (FedAvg). During the training, a selected model from the centralized server is downloaded, and consequently, a fixed number of federated updates that include multiple epochs of loops of updating the selected model to a certain number of rounds using the local data are performed. Afterwards, the selected model that is trained at the end of the federated updates is uploaded back to the centralized server. The federated updates continue until the maximum computation budget is exhausted.

Federated learning, as mentioned, leverages the benefits of machine learning models trained with decentralized data. In the context of connected autonomous vehicle networks, federated learning shares learning from relevant data at the edges of the network with selected centralized servers to minimize sharing raw data, enhancing the performance of the machine learning model utilizing the decentralized network. The intention is to keep the advantage of having the data localized to the edge but allow the aggregation of local inferring of edge-based neural networks. In the threat adaptation application in connected autonomous vehicles, federated learning allows the vehicle networks to collaboratively continue to strengthen the sparse training labels, reducing the uncertainty and missed detections of the autonomous driving threat detection systems.

4.2. Advantages and Limitations

On the other hand, the proposed FL learning-based security solutions have some important limitations. In order to efficiently process participants' data and reduce the number of exchanged model parameters, the training of A/V security algorithms remains within the edge cloud. By excluding such the external A/V security applications, the model never gains a sufficient understanding of the external actors' motives, intentions, and reasoning, which potentially allow the black-box-type adversarial attacks by constraint fulfillment. Consequently, the accompanying theoretical study should necessarily be developed for the proposed FL A/V security algorithms. Furthermore, several aspects of FL for A/V network security functionality need to be more thoroughly investigated before the wide industrial deployment could be considered. First, it is indisputable that FL-based privacy-preserving algorithms remain at the concept level, failing to meet the requirements of the relevant policies, such as GDPR and similar data protection laws. Second, while the federated architecture explicitly limits the exposure of the data at rest, proposals to in-transit data prevention should be better addressed. Third, applying the FL-based A/V cybersecurity algorithms results in the differences in the threat perception range, due to the adjustments according to local data.

The FL approach was proposed as an alternative, or at least a compliment, to a traditional centralized ML approach for A/V security. As it was demonstrated for the realistic network problems, the performance of the FL A/V ML models provided sufficient accuracy for further A/V decision-making support. Although the proposed solutions vary depending on the learning algorithm and AV verified dataset issues, overall, the FL is a conceptually simple approach. It requires only a small set of control parameters and after the predefined setup, the FL training is performed similarly to the non-federated algorithm. In general, the FL-based additive security mechanisms are ensured, which benefits the entire A/V network deployments as the integrity of all cooperative machine learning models is simultaneously preserved, while the independence of the updated A/V local algorithms is vaguely maintained.

5. Collaborative Framework Design

The final output (1-of-K encoding) of the collaborative training process is a distributed ML model, which is subject to the hard quantization errors enforced by the coding and transmission constraints. If the members with the trained ML models have much more confidence in a specific class, the final model may not provide a large enough margin in the decision space, so the small perturbation in the observation may lead to a misclassification with respect to the adjacent classes. Such a challenging nature is more critical when the applications are about the pattern disturbances. For example, for fingerprint identification system, there might be infinite different ways that leaving the fingerprint on the sensor, but a small perturbation might push the result to a mismatch state. Without sharing local models, some works have investigated the distributed ML models with heterogeneous training corpora, and the contributions in this line are about handling cross-domain reforms and sharing a few of model updates under secure computation settings (e.g., compute the privatized gradients or share the model updates).

This paper presents a preliminary study on enabling federated learning towards providing much-needed defenses for sophisticated attackers who compromise the testing corpora and bypass the supervised ML-based detection techniques. In particular, we overview the current issues and highlight the process, system, and cyber-physical challenges to enable an effective federated learning. Under the spotlight after the successful adaption in the smart phone, federated learning becomes more attractive in a collaborative environment that members can train their machine learning models iteratively with the hosted local and/or crowd-sourced data without exchanging an excessive amount of data and inference results. In the traditional supervised machine learning context, "the more data, the better the model" canonical wisdom, members may treat their model (and perhaps their own data) as "private assets" and hesitate to join the federation. Especially for the sensitive corpora, intelligence, and security applications, it may not be possible or willing to share/test the model and final decision for governments, intelligence agencies, or commercial service providers.

5.1. System Architecture

Gaussian Mixture - Out of Memory We use Gaussian Mixture as a widely adopted representation of the data distribution in ADAS applications. For off-the-shelf inference, we implement a Gaussian Mixture - out of Memory FedoraTEE shown in 5 that encodes a number of Gaussian distributions approximating the target Gaussian distribution as a global input tensor to the CPU. At each round, each edge device computes the local Gaussian representation of its own samples, aggregates the mean and count of the Gaussian distributions, sends them to concurrent aggregate routines in the remote edge devices, and retrieves their contributions in reverse order. The edge devices utilizing FedoraTEE conduct distributed GMM updates and exploit model averaging capabilities of output trees within their local inference layer after receiving the cumulative data distribution obtained in this TensorFlow function from another device's population statistics. The aggregate wrapper invokes FedoraTEE-specific serialization and deserialization to support user-defined interfaces.

5.2. Communication Protocols

The Sensor Type List is a binary string of length y: the i-th bit, i ∈ {1, 2, 3, ..., y}, has the value of 0 if the vehicle does not have the i-th sensor data type, or 1 if the vehicle does. The structure of a Sensor Status message is as follows: Let n be the total number of vehicle types, and {S1, S2, ..., Sn} be the sensor types. Each vehicle type broadcasts the Sensor Status message at a time instantaneous after a random back-off period. Upon the receipt of a message, each vehicle will update the Sensor Availability List, which reflects the real-time complete existence of sensor data over the AVN. The structure of the Sensor Availability List is shown in Figure 4.

Autonomous vehicles operate and communicate within a group, which forms an Autonomous Vehicle Network (AVN). An efficient communication protocol is essential to enable real-time data transfer among the vehicles. We describe the message exchange process within the AVN in this section. At the beginning of a communication episode, vehicles broadcast their Sensor Status message to notify each other about the availability of their local data. This is to allow other vehicles to find out if the vehicle carrying data of a certain type is close enough to establish a connection. The structure of the Sensor Status message is composed of the Sensor Type List field to specify the type of sensor data the transmitting vehicle holds, and the Sensor Mask field to denote the relevance of each sensor.

Section 5.2. Communication Protocols

6. Experimental Evaluation

We evaluate Deep Autoencoder, CNN-LSTM, plain FCNN, stacked LSTM, and ResNet architectures using both federated learning and on-device learning on the Joint Edge-Cloud Testbed, which is composed of two Intel NUCs and one Jetson TX2 device. We consider the NUCs to be located on the edge network, and correspondingly the Jetson TX2 is considered to be an MEC device close to the radio access network (RAN). Two NUCs represent the radio core network data centers, and each data center has benign and malicious traffic sent to them from different traffic generators. We perform centralized learning and on-device learning using Keras with a Tensorflow backend. Using the communication infrastructure model proposed in Section 5.1, we derive the cost of communication between edge network elements through a Wi-Fi 6 channel. As the communication cost is a small aspect of this paper's contribution, we do not perform experimental validation on the testbed but instead rely on the model. The results of centralized learning are regarded as ground truth.

In this section, we experimentally evaluate the performance of federated learning and on-device learning for cyber threat detection in vehicular networks. Our evaluation is aimed at assessing the following questions: How well can federated learning and on-device learning approach the detection accuracy of recent centralized view approaches? What is the sensitivity of federated learning and on-device learning to the availability of learning data from different network zones? What is the level of communication cost (i.e., model upload size) of federated learning? As we do not have ground truth threat labels for real-world 5G traffic, our experiments are performed with a network traffic dataset collected at Bell Labs using traffic generators that emulate different network zones. Each network traffic zone corresponds to a data center sending service traffic to end-user devices, and within this traffic, malicious traffic patterns are injected. The benign traffic for each data center corresponds to six distinct network traffic types, including typical user behavior and machine-to-machine (M2M) communication in NG-RAN network slices.

6.1. Datasets and Simulators

6.1. Datasets and Simulators The models are trained using two classes of datasets: 1) Homogeneous datasets: Python packages such as Keras-RL or OpenAI-Gym do not provide dataset collections to benchmark our models. Researchers can use simulators like BitEx, Flow, or SUMO to aggregate a good number of homogeneous auto-generated datasets. Generated datasets can be divided into three categories: 'test', 'validation', and 'training'. Researchers can add different types of noise data to evaluate how resilient a given model is at classifying given situations. Besides, adequate imbalanced datasets may test performance: high rare situations could be stressed in classifying traffic abnormal events. Although class imbalance is a common problem, imbalanced datasets could be barely created by inflating specific traffic situations. 2) Heterogeneous datasets: In addition to the above simulation-based datasets, we use public cyber-physical datasets containing all information state table composed by elements of attack execution such as known types of advanced persistent threats, known types of network assaults, or unusual driving states. Homogeneous datasets can be combined with these datasets and then labeled, after performing exploration regarding different abnormal traffic situations.

In this section, we present the simulation models and training datasets we used to simulate learning data and settings for the three above-mentioned training and evaluating machine learning models.

6.2. Performance Metrics

They are some of the widely used performance metrics used to evaluate the data classification models. There are many more metrics like specificity, area under the curve etc., that can be used to evaluate the model performance.

F-Score (F1-Score): It is the harmonic mean of the precision and recall. The formula to calculate the F-Score is given below.

Recall (Sensitivity): Also known as true positive rate, it measures how many of the actual positive data points were predicted correctly as positive. The formula to calculate the recall is given below.

Precision: Also known as the positive predictive value, it measures how many of the model's positive predictions were actually correct. The formula to calculate the precision is given below.

Accuracy: It measures how many of the actual labels that the model has predicted correctly. The formula to calculate the accuracy is given below.

This section introduces some of the widely used performance metrics used for evaluating the data classification models. If we have an equal number of data points in different classes, then we can evaluate the performance of the classification model using different performance metrics. For example, accuracy is a widely used metric, but it may fail to provide meaningful information if the data is imbalanced. Even though the model has very high accuracy, if the minority class is misclassified significantly, then the accuracy may mislead you. In such cases, confusion matrix based performance metrics will be used generally to evaluate the model performance more effectively.

7. Case Studies

Throughout this paper, two case studies are presented to discuss the challenges, issues, and corresponding resolves for collaborative threat detection in autonomous vehicle networks.

More detailed and integrated deployment and verification of these solutions are the topics of future research. Firstly, trustworthiness-based threat detection specifications and implementation in V2V networks are described and employed to discuss the challenges of distributed event detection, especially the sign of attack detected. Secondly, a neural network knowledge transfer-based convictive results consolidation approach is introduced to sophisticate the trustworthiness collaborative results. Preliminary verification from preliminary trustworthiness-enhanced detection performance and the resulting risk exhibits the potential effectiveness of the solution. More detailed modeling and feasibility experimental approaches and works for the comprehensive deployment of incident detection with robustness in autonomous vehicle networks are ongoing research in our group.

Two collaborative threat detection approaches are implemented by incorporating distributed learning models into all-connected V2V networks. First, a trustworthiness-collaborative event detection prototype is developed by a cellular automata, and a deep learning model is used to accelerate the decision-making process for malicious event detection. Moreover, by incorporating different trustworthiness mechanisms in threat detection, neural network knowledge transfer-based convictive results consolidation approach invites multiple distributed event detention modules for collective learning, and stretch-based knowledge transfer is used to enhance results reliability by modeling outlier event detection confidence. Preliminary detection performance under different leveraging, mobility, and common attack scenarios was verified through simulations and simplified feasibility experiments, whereas plausible risk is outlined by considering attack-induced plateau changes in traffic pattern and future work is highlighted to implement robust threat detection in edge computing environments.

7.1. Real-World Applications

The approaches described in this paper should foster new applications of collaborative, secure, and scalable model training in real-world use cases to identify possible vectors of attack in autonomous car wireless networks using privacy-preserving machine learning algorithms. Enterprises or organizations in the financial sector and FinTech could form alliances in order to detect threats or compliance risks through machine learning without merging their customer data or exposing it to a central third party. With the increasing need for privacy for individual car owners or insurance companies, alternative methodologies that guarantee privacy of training data will be necessary to foster widespread use of ML techniques to guarantee safe and liability-aware driving experiences.

Federated learning has been a very promising learning approach for AI applications that have strict constraints on the privacy of training data, such as healthcare, finance, etc. Yet, it has not been studied in detail in the CV domain. This study proposes several FL methodologies that can be applied in the CV field. Several real-world implementations and applications of FL could find widespread use in privacy-preserving machine learning applications for collaborative learning and training across multiple edge devices. Enterprises may want to use machine learning integrations at the edge for several use cases for privacy or security reasons, and FL can help to preserve the data while increasing the accuracy of the model.

8. Security and Privacy Considerations

Within the training data, compromising the personal information of AV drivers via the shared training set could have lasting consequences if breaches occur. Leakage of training data could provide a malicious actor information that could be exploited to attack drivers or undermine a fleet of AVs, therefore endangering the collective safety of on-the-road vehicles. In the shared information communication network, adversaries could potentially eavesdrop or manipulate transmissions to interfere with the federated learning model. Machine learning models themselves are also vulnerable. Unique samples from the training set might be inferred from model updates, and mitigation of federated adversaries from poisoning the shared model or hijacking the targeted learning task threats are paramount. Secure aggregation, cryptographic protocols, differential privacy, and homomorphic encryption could provide solutions to such threats.

The security and privacy considerations of threat detection in AVs are paramount to the successful deployment of collaborative ML models. Understanding and addressing potential vulnerabilities in privacy of the training data, communication network, and ML model is critical in protecting the collective safety of all AVs utilizing collaborative threat detection in a federated learning environment. Certain training scenarios could pose a greater risk to datasets, wherein modeling of these threats could illuminate potential areas of security research.

8.1. Threat Models

The autonomous vehicle is a networked system that can cooperate with other entities, including ground traffic and air traffic control. There are various threats that can exploit vulnerabilities of these networking interfaces, leading to potentially catastrophic consequences, such as accidents. Some of these threats may be caused by intelligent adversaries, and continuously updated models are needed to maintain awareness and fight back. Resolving these threats, however, in a system of vehicles not having continuous connectivity due to lossy channels and dynamically changing network topologies, necessitates the sharing of sensitive data, such as driving commands and video feeds. Therefore, any solution must maintain a balance between its learning capacity and preserving the local privacy of its entities.

Federated learning (FL) allows many agents to collaboratively learn models, such as centralized ones, without sharing their local data. This capability makes it an attractive option to collaboratively develop threat detection models for autonomous vehicle networks. In this chapter, we analyze both horizontal and vertical FL approaches to enable threat detection in centralized, hybrid, and full-federated architectures. We then compare these approaches in terms of detection performance and requirements.

8.2. Privacy-Preserving Mechanisms

The federated learning initiates the sharing of the initial weights alone of the collaborative model by the fleet manager with all vehicles in the fleet. The combined vehicle features are used by homomorphic encryption to make the vehicle identity and traffic scenarios oblivious, thus preventing the revelation of threatening situations. United training data from vehicles with different traffic scenarios but with the same combination of vehicles can weaken the threat detection model in the sharing stage. These cryptic queries are submitted to the fleet manager and invoked to request the same shared encrypted weights. Though the decryption for these homomorphic queries could occur in a secure hardware enclave only inside the fleet manager, activating such a number of queries could degrade the performance of the secure enclave over time. The relevant information about these cryptic queries is revealed and aggregated together, putting an upper bound on liability insurers' interests. It's shareable access to the models that would approximate the homomorphic queries encrypted and allow access to the model.

Federated learning, used for training the collaborative threat detection model in this research, does not involve any exchange of raw data with the destination on the network until the shared threat detection model weight updates are available. The collaborative model is built with the goals of minimizing communication overhead, meeting tolerance to communication failures, being energy-efficient, having minimum synchronization overheads for the training, and preserving the privacy of the vehicles in the network. Privacy-preserving mechanisms for the federated model approach used in this research are homomorphic encryption, exclusively disjoint record shuffling, and mocking the real data with crafted synthetic data. Data is utilized, shared, and updated among vehicles and the fleet manager in a homomorphic-encrypted fashion, enabling such dependency structures in the cryptographic domain itself. Only the fleet manager has the authority to decrypt and access all shared data from vehicles.

9. Future Directions

One of the primary sources of threat model diversity is the potential issue of differing event rates across a wide range of real-world driving environments. Extensive simulation capabilities could be created to encompass representative threat environments. These could then be used to scale real-world event rates to assess the impact of the new infrastructure's threat detection effectiveness under different global learning data accumulation rates. The above infrastructure also describes both point-in-time online threat analysis, as well as methods allowing multiple vehicles to collaborate in the updating of their own-background event distributions due to the infinite delays that characterize real-time learning infrastructure. This federation can occur via the use of a central processing point, or by the exchange of information packets that characterize the worlds that individual vehicles' various threat models operate within.

The proposed federated learning infrastructure leverages the character of novel autonomous vehicle environments to achieve effective threat detection without compromising individual vehicle operational constraints. It does this by allowing individual vehicles to retain full control of the more expensive end-of-route threat detection process, while creating a federated environment where more efficient temporal and spatial threat identification can be learned via plural threat models and the sharing of local experience acquired based on similar routes at different vehicle operational instances. This infrastructure thus enables efficient threat identification through the leveraging of spatial and temporal independence. Future research directions include (i) the effect of real-world event rates, (ii) federated threat model updating, and (iii) feature selection.

9.1. Emerging Technologies

At a lower level of the data-to-decision spectrum, where real-world tactical responses to threats in infrastructure systems that can be passively monitored are desired, emerging technologies supporting software-defined networking based on an edge cloud may play a role. The ability to simulate realistic system states, while the legitimate cyber-physical control software responds as it would in the real world, by replaying actual system sensor data would enable testing of these myriad system states, including those not seen or considered in the system's original development, against a broad assortment of development and machine learning based security techniques that drive the future of secure autonomy.

Future cyber threats may also be quite different from current attackers employing brute force to enter networks. Sophisticated nation-states may develop high-fidelity simulations of cities within which they can train autonomous vehicles on identifying safe and insecure areas, all through subtle alteration of the testing data leading to concerted action by the vehicles. Depending on how these threats are conceived, federated learning may address part of the problem, though some threats would not be amenable to this approach. Federated learning is designed for optimization tasks where distributed input and computation from multiple data owners exist, and a shared model is to be learned or refined. It might be of use as a privacy-preserving technique to build intelligence in a more secure manner, safely sharing only learned information—parameter updates—rather than raw data.

9.2. Research Challenges

To enable a tridimensionality to the task of threat detection, the local intelligence solution participates as the first feature-extraction layer, detecting local threats. This is vital for establishing a self-organizing communication network. The second layer is responsible for collecting information from the feature-extraction layer of each node and determining the threats inside the cooperative environment. There are many challenges that need to be addressed for the successful deployment of the cooperative threat-reasoning federated learning approach. This section discusses some of the major challenges encountered and the research around the topic. These challenges are: sharing the most relevant features given the constraints of the vehicle network with different missions; the high privacy threat states of sessions given the payload content and location of the data set; global proposals to make the presence of threats appear in the federated nodes at the end of the feature-extraction layer.

Centralized methods work by transferring all data collected from a client's device to the cloud server. Unfortunately, this raises privacy concerns due to the exposure of user data at the server. Federated learning is an approach that trains a system across multiple devices or servers holding local data samples by deleting raw data from devices prior to communicating parameter updates with one another. This solves the issues of centralized learning by learning it in the same location where the learned function will be used. However, the task of developing an AI system that can work in a collaborative environment of vehicles that share cooperation to detect possible threats is daunting. This multi-layer process is performed within the vehicle network for the purpose of the cooperative threat detection system. This is due to the intermittent connection and heterogeneous architectures of the vehicles while inside the network.

10. Conclusion

Federated learning approaches have attracted significant attention in many fields, as it enables on-device intelligence while avoiding personal data transmitting concerns. Similarly, in the autonomous driving setting, federated learning empowers the participating parties to collaboratively construct a threat detection model without sharing their driving data directly. To design a successful collaborative threat detection model in the federated learning environment, achieving high learning performance and sustaining communication efficiency is challenging. In this work, we have studied how to address these challenges by proposing a unified and practical method for efficient fleet-scale model training and performing a systematic evaluation of federated learning performance in real vehicle networks.

In this work, we study federated learning approaches for collaborative distributed threat detection in autonomous vehicle networks. In contrast to conventional standalone federated learning models, the proposed federated learning approaches consistently adapt and improve performance across different fleet collaborations and systematically balance learning performance and communication efficiency through optimization of device selection. Our simulation results show the proposed method delivers 13% higher collaborative learning performance compared to the benchmark approach. At the same time, the proposed method saves over 90% of computing resource spent on model aggregation, leading to 5X faster convergence speed. The proposed method generates consistent improvements over the benchmark approach as varying network scale and vehicle performance constraints. In addition, the performance lower bound is presented so that the convergence speed and global model accuracy can be carefully balanced based on the specifications of actual vehicle networks.

Reference:

1. Perumalsamy, Jegatheeswari, Bhargav Kumar Konidena, and Bhavani Krothapalli. "AI-Driven Risk Modeling in Life Insurance: Advanced Techniques for Mortality and Longevity Prediction." *Journal of Artificial Intelligence Research and Applications* 3.2 (2023): 392-422.

2. Karamthulla, Musarath Jahan, et al. "From Theory to Practice: Implementing AI Technologies in Project Management." *International Journal for Multidisciplinary Research* 6.2 (2024): 1-11.

3. Jeyaraman, J., Krishnamoorthy, G., Konidena, B. K., & Sistla, S. M. K. (2024). Machine Learning for Demand Forecasting in Manufacturing. *International Journal for Multidisciplinary Research*, *6*(1), 1-115.

4. Karamthulla, Musarath Jahan, et al. "Navigating the Future: AI-Driven Project Management in the Digital Era." *International Journal for Multidisciplinary Research* 6.2 (2024): 1-11.

5. Karamthulla, M. J., Prakash, S., Tadimarri, A., & Tomar, M. (2024). Efficiency Unleashed: Harnessing AI for Agile Project Management. *International Journal For Multidisciplinary Research*, *6*(2), 1-13.

6. Jeyaraman, Jawaharbabu, Jesu Narkarunai Arasu Malaiyappan, and Sai Mani Krishna Sistla. "Advancements in Reinforcement Learning Algorithms for Autonomous Systems." *International Journal of Innovative Science and Research Technology (IJISRT)* 9.3 (2024): 1941-1946.

7. Jangoan, Suhas, Gowrisankar Krishnamoorthy, and Jesu Narkarunai Arasu Malaiyappan. "Predictive Maintenance using Machine Learning in Industrial IoT." *International Journal of Innovative Science and Research Technology (IJISRT)* 9.3 (2024): 1909-1915.

8. Jangoan, Suhas, et al. "Demystifying Explainable AI: Understanding, Transparency, and Trust." *International Journal For Multidisciplinary Research* 6.2 (2024): 1-13.

9. Krishnamoorthy, Gowrisankar, et al. "Enhancing Worker Safety in Manufacturing with IoT and ML." *International Journal For Multidisciplinary Research* 6.1 (2024): 1-11.

10. Perumalsamy, Jegatheeswari, Muthukrishnan Muthusubramanian, and Lavanya Shanmugam. "Machine Learning Applications in Actuarial Product Development: Enhancing Pricing and Risk Assessment." *Journal of Science & Technology* 4.4 (2023): 34-65.