Securing Connected & Autonomous Vehicles: Challenges Posed by Adversarial Machine Learning and the Way Forward

Adnan Qayyum, Muhammad Usama, Junaid Qadir, *Senior Member, IEEE*, and Ala Al-Fuqaha, *Senior Member, IEEE*

***Abstract*—Connected and autonomous vehicles (CAVs) will form the backbone of future next-generation intelligent transportation systems (ITS) providing travel comfort, road safety, along with a number of value-added services. Such a transformation—which will be fuelled by concomitant advances in technologies for machine learning (ML) and wireless communications—will enable a future vehicular ecosystem that is better featured and more efficient. However, there are lurking security problems related to the use of ML in such a critical setting where an incorrect ML decision may not only be a nuisance but can lead to loss of precious lives. In this paper, we present an in-depth overview of the various challenges associated with the application of ML in vehicular networks. In addition, we formulate the ML pipeline of CAVs and present various potential security issues associated with the adoption of ML methods. In particular, we focus on the perspective of adversarial ML attacks on CAVs and outline a solution to defend against adversarial attacks in multiple settings.**

***Index Terms*—Connected and autonomous vehicles, machine/deep learning, adversarial machine learning, adversarial perturbation, perturbation detection, and robust machine learning.**

I. Introduction

IN RECENT years, connected and autonomous vehicles (CAVs) have emerged as a promising area of research. The connected vehicles are an important component of intelligent transportation systems (ITS) in which vehicles communicate with each other and with communications infrastructure to exchange safety messages and other critical information (e.g., traffic and road conditions). One of the main driving force for CAVs is the advancement of machine learning (ML) methods, particularly deep learning (DL), that are used for decision making at different levels. Unlike conventional connected vehicles, the autonomous (self-driving) vehicles have two important characteristics; namely, automation capability and cooperation (connectivity) [1]. In future smart cities, CAVs are expected to have a profound impact on the vehicular ecosystem and society.

The phenomenon of connected vehicles is realized through technology known as vehicular networks or vehicular ad-hoc networks (VANETs) [2]. Over the years, various configurations of connected vehicles have been developed including the use of dedicated short-range communications (DSRC) in the United States and ITS-G5 in Europe based on the IEEE 802.11p standard. However, a recent study [3] has shown many limitations of such systems such as (1) short-lived infrastructure-to-vehicle (I2V) connection, (2) non-guaranteed quality of service (QoS), and (3) unbounded channel access delay, etc. To address such limitations, the 3rd generation partnership project (3GPP) has been initiated with a focus on leveraging the high penetration rate of long term evolution (LTE) and 5G cellular networks to support vehicle-to-everything (V2X) services [4]. The purpose of developing V2X technology is to enable the communication between all entities encountered in the road environment including vehicles, communications infrastructure, pedestrians, cycles, etc.

The impressive ability of ML/DL to leverage increasingly accessible data, along with the advancement in other concomitant technologies (such as wireless communications), seems to be set to enable autonomous and self-organizing connected vehicles in the future. In addition, future vehicular networks will evolve from normal to autonomous vehicles and will enable ubiquitous Internet access on vehicles. ML will have a predominant role in building the perception system of autonomous and semi-autonomous connected vehicles.

Despite the development of different configurations of connected vehicles, they are still vulnerable to various security issues and there are various automotive attack surfaces that can be exploited [5]. The threat is getting worse with the development of fully autonomous vehicles. As the autonomous vehicles are being equipped with many sensors such as cameras, RADAR, LIDAR, and mechanical control units, etc. These sensors share critical sensory information with onboard devices through CAN bus and with other nearby vehicles as well. The backbone of self-driving vehicles is the onboard intelligent processing capabilities using the data collected through the sensory system. This data can be used for many other purposes, e.g., getting information about vehicle kinetics,

Manuscript received May 30, 2019; revised October 31, 2019; accepted February 9, 2020. Date of publication February 19, 2020; date of current version May 28, 2020. *(Corresponding author: Adnan Qayyum.)*

Adnan Qayyum is with the Computer Science Department, Information Technology University, Lahore 54000, Pakistan (e-mail: adnan.qayyum@itu.edu.pk).

Muhammad Usama, and Junaid Qadir are with the Electrical Engineering, Information Technology University, Lahore 54000, Pakistan.

Ala Al-Fuqaha is with the Information and Computing Technology Division, College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar, and also with the Computer Science Department, College of Engineering and Applied Sciences, Western Michigan University, Kalamazoo, MI 49008 USA.

Digital Object Identifier 10.1109/COMST.2020.2975048

TABLE I
COMPARISON OF THIS PAPER WITH EXISTING SURVEY AND REVIEW PAPERS ON THE SECURITY OF MACHINE LEARNING (ML) AND CONNECTED AND AUTONOMOUS VEHICLES (CAVs). (LEGEND: $\surd$ MEANS COVERED; $\times$ MEANS NOT COVERED; $\approx$ MEANS PARTIALLY COVERED)

Year	Authors	Publisher	Papers Cited	Focused Area	Conventional Challenges	Threat Models	Adversarial ML	Robust ML Solutions	Autonomous Vehicles	Connected Vehicles	Open Research Issues
2014	Mejri et al. [10]	Elsevier Vehicular Communication	69	Security of vehicular networks	$\surd$	$\times$	$\times$	$\times$	$\times$	$\surd$	$\times$
2016	Gardiner et al. [11]	ACM Computing Surveys (CSUR)	40	Security of ML for malware classification	$\times$	$\surd$	$\surd$	$\approx$	$\times$	$\times$	$\times$
2018	Chakraborty et al. [12]	arXiv	79	Adversarial attacks and defenses	$\times$	$\surd$	$\surd$	$\surd$	$\times$	$\times$	$\times$
2018	Akhter et al. [13]	IEEE Access	195	Adversarial attacks and defenses in computer vision	$\times$	$\times$	$\surd$	$\surd$	$\approx$	$\times$	$\times$
2018	Siegel et al. [14]	IEEE Transactions on ITS	198	Survey on connected vehicles' landscape	$\surd$	$\times$	$\times$	$\times$	$\times$	$\surd$	$\times$
2018	Hussain et al. [2]	IEEE Communications Surveys and Tutorials (COMST)	230	Autonomous cars: research results, issues and future challenges	$\surd$	$\times$	$\times$	$\times$	$\surd$	$\times$	$\surd$
2019	Yuan et al. [15]	IEEE Transactions on NN & LS (TNNLS)	146	Adversarial attacks and defenses for deep learning systems	$\times$	$\surd$	$\surd$	$\surd$	$\times$	$\times$	$\times$
2019	Wang et al. [16]	arXiv	128	Adversarial ML attacks and defenses in text domain	$\times$	$\surd$	$\surd$	$\surd$	$\times$	$\times$	$\times$
2019	Our Paper		239	Security of CAVs and ML	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$

traffic flow, road, and network conditions, etc. Such data can be potentially used for improving the performance of the vehicular ecosystem using adaptive data-driven decision making and can also be used to accomplish various destructive objectives. Therefore, ensuring data integrity and security are necessarily important to avoid various risks and attacks on CAVs.

It is common for the perception and control systems of CAVs to be built using ML/DL methods. However, ML/DL techniques have been recently found vulnerable to carefully crafted adversarial perturbations [6] and different physical world attacks have been successfully performed on the vision system of autonomous cars [7], [8]. This has raised many privacy and security concerns about the use of such methods particularly for security-critical applications like CAVs. In this paper, we aim to highlight various security issues associated with the use of ML and we present a review of adversarial ML literature mainly focusing on CAVs. In addition, we also present a taxonomy of possible solutions to restrict adversarial ML attacks and open research issues on autonomous, connected vehicles, and ML.

ML in general and DL schemes specifically perform exceptionally well in learning hidden patterns from data. DL schemes such as deep neural networks (DNN) have outperformed human-level intelligence in many perception and detection tasks by accurately learning from a large corpus of training data and classifying/predicting with high accuracy on unseen real-world test examples. As DL schemes produce outstanding results, they have been used in many real-world security-sensitive tasks such as perception system in self-driving cars, anomaly and intrusion detection in vehicular networks, etc. ML/DL schemes are designed for benign and stationary environments where it is assumed that the training and test data belongs to the same statistical distribution. The application of this assumption in a real-world application is flawed as training and test data can have different statistical distributions which gives rise to an opening for adversaries to compromise the ML/DL-based systems. Furthermore, the lack of interpretability of the learning process, imperfections in training process, and discontinuity in the input-output relationship of DL schemes also resulted in an incentive for adversaries to fool the deployed ML/DL system [9].

Contributions of This Paper: In this paper, we build upon the existing literature available on CAVs and present a comprehensive review of that literature. A comparison of this paper with existing surveys on security of CAVs is presented in Table I. The following are the major contributions of this study.

1) We formulate the ML pipeline of CAVs and describe in detail various security challenges that arise with the increasing adoption of ML techniques in CAVs, specifically emphasizing the challenges posed by adversarial ML;
2) We present a taxonomy of various threat models and highlight the generalization of attack surfaces for general ML, autonomous, and connected vehicle applications;
3) We review existing adversarial ML attacks with a special emphasis on their relevance for CAVs;
4) We review robust ML approaches and provide a taxonomy of these approaches with a special emphasis on their relevance for CAVs; and
5) Finally, we highlight various open research problems that require further investigation.

Organization of the Paper: The organization of this paper is depicted in Figure 1. The history, introduction, and various challenges associated with connected and automated vehicles (CAVs) are presented in Section II. Section III presents an overview of the ML pipeline in CAVs. The detailed overview of adversarial ML and its threats for CAVs are described in Section IV. An outline of various solutions to robustify applications of ML along with common methods and recommendations for evaluating robustness are presented in Section V. Section VI presents open research problems on the use of ML in the context of CAVs. Finally, we conclude the paper in Section VII. A summary of the salient acronyms used in this paper is presented in Table II for convenience.

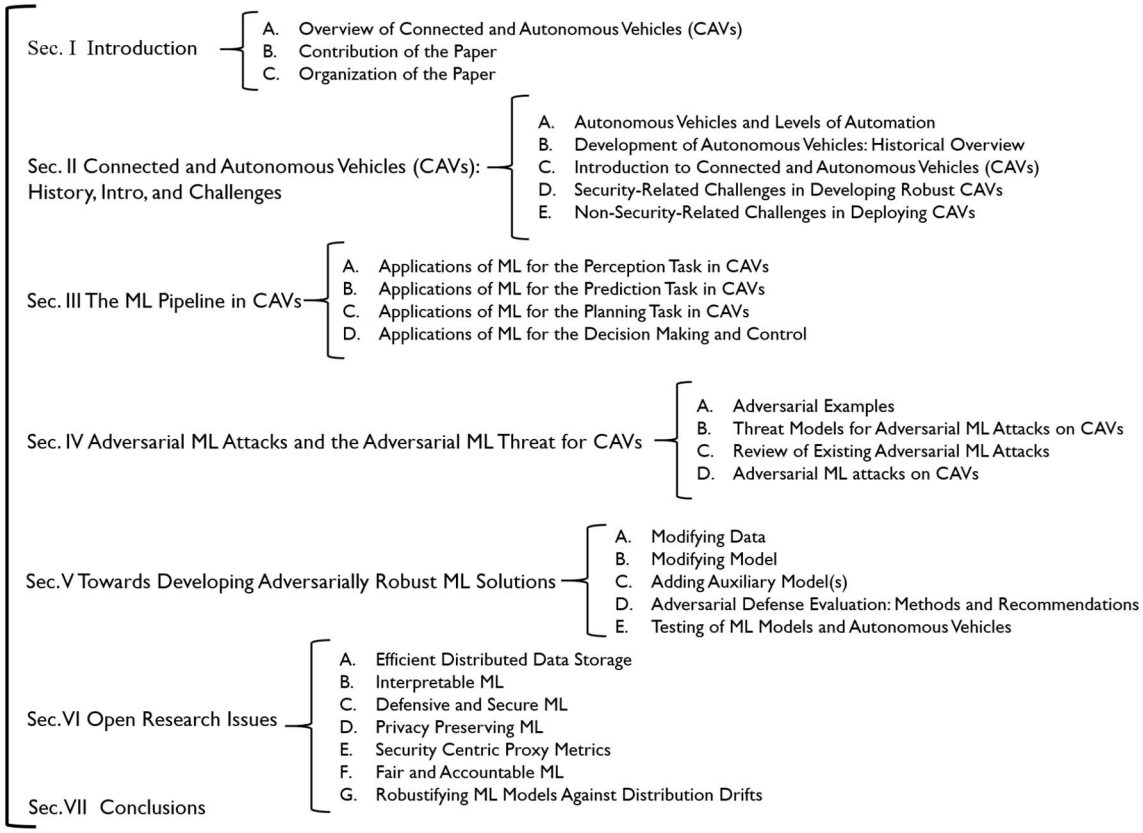

Fig. 1. Outline of the paper.

II. Connected and Autonomous Vehicles (CAVs): History, Introduction, and Challenges

In this section, we provide the history, introduction, background of CAVs along with different conventional and security challenges associated with them.

A. Autonomous Vehicles and Levels of Automation

The society of automotive engineers (SAE) has defined a taxonomy of driving automation that is organized in six levels. The SAE defined the potential of driving automation at each level that is depicted in Figure 2. Moreover, according to a recent scientometric and bibliometric review article on autonomous vehicles [17], different naming conventions have been used over the years to refer to autonomous vehicles. These names are illustrated in Figure 3; note that the year denotes the publication year of the first paper mentioning the corresponding name.

The SAE defines the operational design domain (ODD) for the safe operation of autonomous vehicles as "the specific conditions under which a given driving automation system or feature thereof is designed to function, including, but not limited to, driving modes" [18]. ODD refers to the domain of operation which an autonomous vehicle has to deal with. An ODD representing an ability to drive in good weather conditions is quite different from an ODD that embraces all kinds of weather and lighting conditions. The SAE recommends that ODD should be monitored at run-time to gauge if the autonomous vehicle is in a situation that it was designed to safely handle.

B. Development of Autonomous Vehicles: Historical Overview

Self-driving vehicles, especially ones considering lower levels of automation (referring to the taxonomy of automation as presented in Figure 2), have existed for a long time. In 1925, Francis Udina presented a remote controlled car famously known as American wonder. In the 1939-1940 New York World's Fair, General Motors Futurama exhibited aspects of what we call self-driving car today. General Motors and RCA initiated the first work around the design and development of self-driving vehicles in the early 1950s [19] that was followed by Prof. Robert Fenton at The Ohio State University from 1964–80.

In 1986, Ernst Dickens at University of Munich designed a robotic van that can drive autonomously without traffic and by 1987 the robotic van drove up to 60 Km/hr. Later he started the development of driving scenes recognition tools using video imagery [20] that was followed by a demonstration performed under the Eureka Prometheus project. The super-smart vehicle systems (SSVS) program in Europe [21] and Japan [22] were also based on the earlier work of Ernst Dickens. In 1992, four vehicles drove in a convoy using magnetic markers on the road for relative positioning, a similar test was repeated in 1997 with eight vehicles using radar systems and

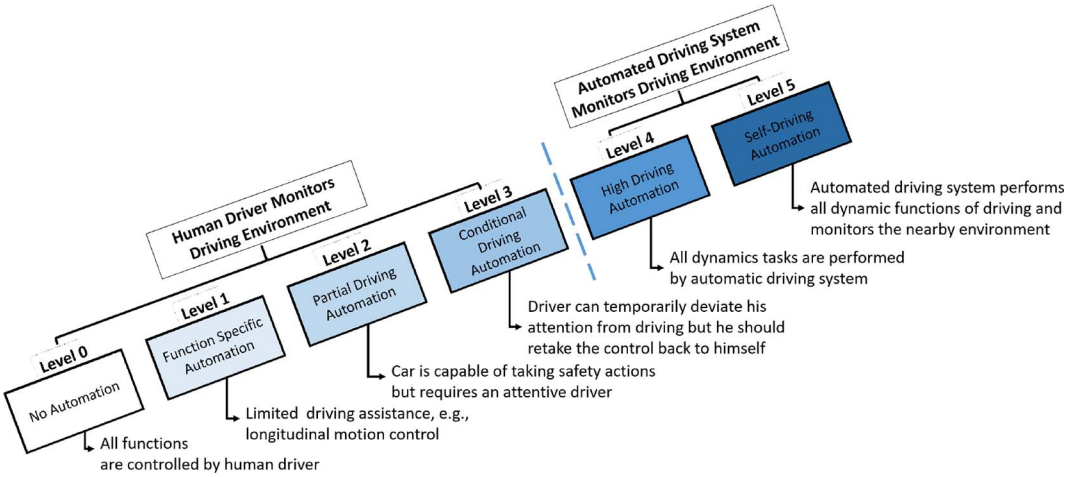

Fig. 2. The taxonomy of the levels of automation in driving.

TABLE II
LIST OF ACRONYMS

BSM	Basic Safety Message
BFGS	Broyden–Fletcher–Goldfarb–Shanno Algorithm
CAN	Controller Area Network
CAVs	Connected and Automated (Autonomous) Vehicles
CIFAR	Canadian Institute for Advanced Research
CNN	Convolutional Neural Network
C&W	Carlini and Wagner Algorithm
DARPA	Defense Advanced Research Projects Agency
DL	Deep Learning
DNN	Deep Neural Network
ECUs	Electronic Control Units
FGSM	Fast Gradient Sign Method
GAN	Generative Adversarial Networks
GPS	Global Positioning System
GTSDB	German Traffic Sign Detection Benchmark
GTSRB	German Traffic Sign Recognition Benchmark
IoV	Internet of Vehicles
JSMA	Jacobian-based Saliency Map Attack
L-BFGS	Limited-memory BFGS
LIDAR	LIght Detection and Ranging
LISA	Laboratory for Intelligent & Safe Automobiles
LSTM	Long Short-Term Memory
MC/DC	Modified Condition/Decision Coverage
ML	Machine Learning
MNIST	Modified National Institute of Standards and Technology
ODD	Operational Design Domain
RADAR	RAdio Detection And Ranging
RL	Reinforcement Learning
RSU	Road-Side Unit
SAE	Society of Automotive Engineers
SVM	Support Vector Machine
VANETs	Vehicular Ad-hoc Networks
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
V2X	Vehicle to Everything
VGG	(Oxford University's) Visual Geometry Group
YOLO	You Only Look Once (Classifier)

V2V communications. This work has paved the way for modern adaptive cruise control and automated emergency braking systems. This R&D work then witnessed initiatives of programs like the PATH Program by Caltrans and the University of California in 1986, in particular, the work on self-driving got huge popularity with the demonstration of research work done by the national automated highway systems consortium (NAHSC) during 1994-98 [23] and this climax remained until 2003.

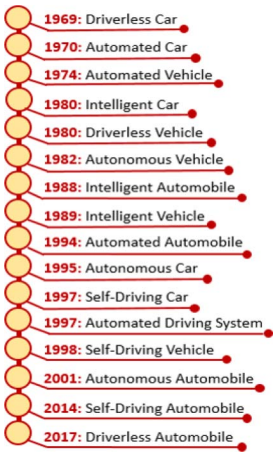

Fig. 3. The illustration of different naming conventions used for referring autonomous vehicles in past years, the year denotes the publication year of first paper mentioning corresponding name. We see that self-driving car is not entirely a new concept and it is referred to through a number of terms. (Source: [17]).

In 2002, the defense advanced research project agency (DARPA) announced the grand autonomous vehicles challenge and held the first episode in 2004. The first grand challenge was won by Carnegie Mellon University (CMU) and their car only drove nearly seven miles where the finish line was at 140 miles. In 2005, the second episode of the DARPA grand challenge was held in which five out of twenty-three teams were able to reach the finish line.

This time Stanford University's vehicle "Stanley" has won the challenge. In the third episode of DARPA grand challenge in 2007, universities were invited to present the autonomous vehicles on busy roads to shift the perception of the public, tech, and automobile industries about the design and feasibility of autonomous vehicles.

In 2007, Google hired the team leads of Stanford and CMU autonomous vehicle projects and started pushing towards their self-driving car design on the public roads. By the year 2010, Google's self-driving car has navigated approximately 140 thousand miles on the roads of California in quest of achieving the target of 10 million miles by 2020. In 2013, VisLab

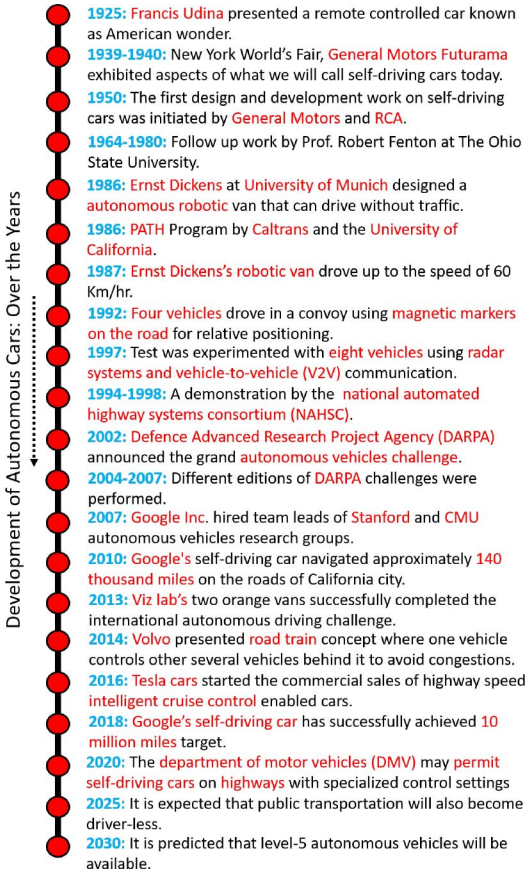

Fig. 4. The timeline for the development of autonomous vehicles.

(a spin-off company of the University of Parma) successfully completed the international autonomous driving challenge by driving two orange vans 15000 km with minimal driver interventions from University of Parma in Italy to Shanghai in China. A year later in 2014, Volvo demonstrated the road train concept where one vehicle controls several other vehicles behind it in order to avoid road congestion. In 2016, Tesla cars have started the commercial sales of highway speed intelligent cruise control based cars with minimal human intervention.

In October 2018, Google self-driving car has successfully achieved the 10 million miles target. It is expected that by 2020 the state departments of motor vehicles (DMV) may permit self-driving cars on the highways with their special lanes and control settings. By 2025, it is expected that public transportation will also become driver-less and by 2030 it is foresighted that we will have level-5 autonomous vehicles.[1] A timeline for the development of autonomous vehicles over the past decades is depicted in Figure 4.

C. Introduction to Connected and Autonomous Vehicles (CAVs)

The term connected vehicles refers to the technologies, services, and applications that together enable inter-vehicles connectivity. In connected vehicles' settings, the vehicles are

[1]https://www.forbes.com/sites/joannmuller/2015/10/15/the-road-to-self-driving-cars-a-timeline/#6731c6f17795.

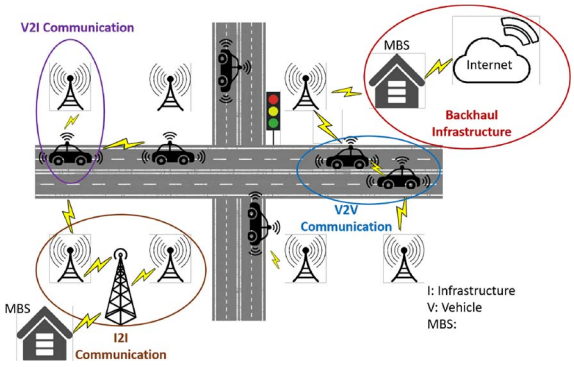

Fig. 5. The basic system architecture of connected vehicles having three types of communications: vehicle-to-vehicle (V2V), infrastructure-to-infrastructure (I2I), and infrastructure-to-vehicle (I2V).

equipped with a wide variety of onboard sensors that communicate with each other via CAN bus and nearby communication infrastructures and vehicles (as illustrated in Figure 5). The applications of connected vehicles include everything from traffic safety, roadside assistance, infotainment, efficiency, telematics, and remote diagnostics to autonomous vehicles and GPS. In general, the connected vehicles can be regarded as a cooperative intelligent transport system [24] and fundamental component of the Internet of vehicles (IoV) [25]. A review of truck platooning automation projects formulating the settings of connected vehicles (described earlier) together with various sensors (i.e., RADAR, LIDAR, localization, laser scanners, etc.) and computer vision techniques is presented in [26]. The key purpose of initiating and investigating such projects is to reduce energy consumption and personnel costs by automated operation of following vehicles. Furthermore, it has been suggested in the literature that throughput on urban roads can be doubled using vehicle platooning [27].

CAVs is an emerging area of research that is drawing substantial attention from both academia and industry. The idea of connected vehicles has been conceptualized to enable inter-vehicle communications to provide better traffic flow, road safety, and greener vehicular environment while reducing fuel consumption and travel cost. There are two types of nodes in a network of connected vehicles: (1) vehicles having onboard units (OBUs), and (2) roadside wireless communication infrastructure or roadside units (RSUs). The basic configuration of a vehicular network is shown in Figure 5. There are three modes of communications in such networks: vehicle-to-vehicle (V2V), infrastructure-to-infrastructure (I2I), and vehicle-to-infrastructure (V2I). Besides these, there are two more types of communication—vehicle to pedestrian (V2P) and vehicle to anything (V2X)—that are expected to become part of the future connected vehicular ecosystem.

In modern vehicles, self-contained embedded systems called electronic control units (ECUs) are used to digitally control a heterogeneous combination of components (such as brakes, lighting, entertainment, and drivetrain/powertrain, etc.) [29]. There are more than 100 such embedded ECUs in a car that are executing about 100 million expressions of code and are interconnected to control and provide different functionalities

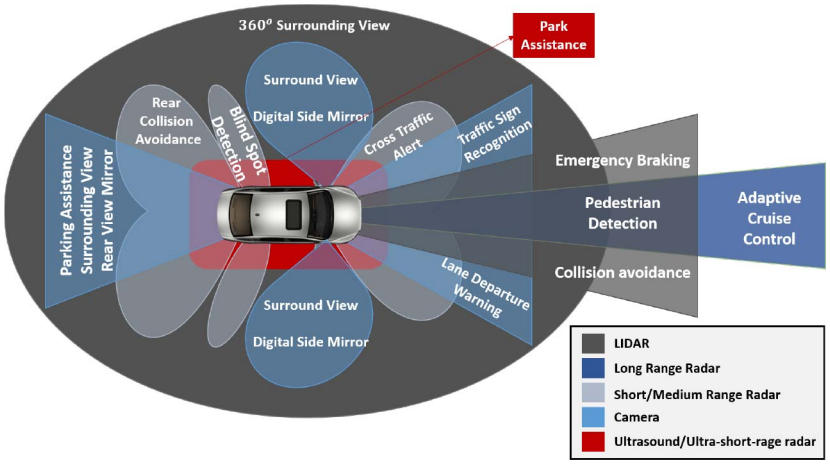

Fig. 6. Autonomous vehicle's major sensor types, their range, and position (figure adapted from [28]).

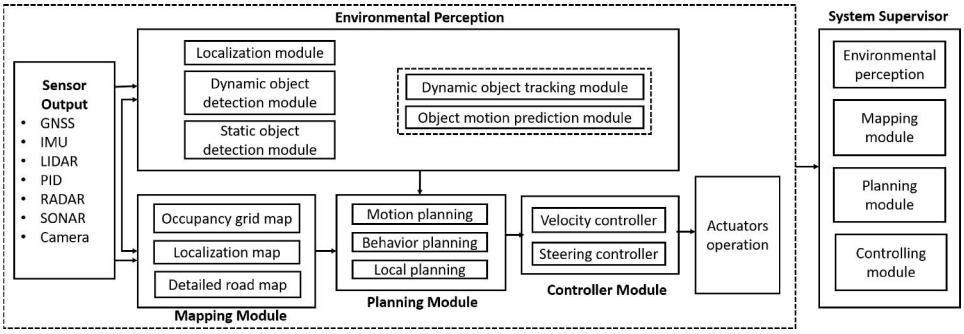

Fig. 7. The systematic software workflow of autonomous vehicles. Nuts and bolts of all important operational blocks of software workflows are depicted to provide reader with a better understanding of the system design involved in developing a state-of-art autonomous vehicle.

such as acceleration, steering, and braking [30]. The security of ECUs can be compromised and remote attacks can be realized to gain control of the vehicle as illustrated in [29].

Modern CAVs utilize a number of onboard sensors including proximity, short, middle, and long range sensors. While each of these sensors works in its dedicated range, they can act together to detect objects and obstacles over a wide range. The major types of sensors deployed in autonomous vehicles and their sensing range are shown in Figure 6 and are briefly discussed next.

- *Proximity Sensors (5m):* Ultrasonic sensors are proximity sensors that are designed to detect nearby obstacles when the car is moving at a low speed, especially, they provide parking assistance.
- *Short Range Sensors (30m):* There are two types of short-range sensors: (1) forward and backward cameras and (2) short range radars (SRR). Forward cameras assist in traffic signs recognition and lane departure while backward cameras provide parking assistance and SRR help in blind spot detection and cross traffic alert.
- *Medium Range Sensors (80-160m):* The LIDAR and medium-range radars (MRR) are designed with a medium range and are used for pedestrian detection and collision avoidance.
- *Long Range Sensors (250m):* Long range radars (LRR) enable adaptive cruise control (ACC) at high speeds in conjunction with the information collected from internal sensors and from other vehicles and nearby RSU [31].

The software design of autonomous vehicles utilizing ML/DL schemes is divided into five inter-connected modules; namely, *environmental perception, mapping module, planning module, controller module,* and *system supervisor*. Figure 7 highlights the software design of autonomous vehicles and it also provides the sensory input required for each software module to perform the designated task.

D. Security-Related Challenges in Developing Robust CAVs

Modern vehicles are controlled by complex distributed computer systems comprising millions of lines of code executing on tens of heterogeneous processors with rich connectivity provided by internal networks (e.g., CAN) [5]. While this structure has offered significant efficiency, safety, and cost benefits, it has also created the opportunity for new attacks. Ensuring the integrity and security of vehicular systems is crucial, as they are intended to provide road safety and are essentially life critical.

Different types of attacks on vehicular networks are described below.

1) Application Layer Attacks: The application layer attacks affect the functionality of a specific vehicular application such as beaconing and message spoofing. Application layer attacks

can be broadly classified as integrity or authenticity attacks and are briefly described below.

(a) *Integrity Attacks:* In the *message fabrication* attack, the adversary continuously listens to the wireless medium and upon receiving each message, fabricates its content accordingly and rebroadcasts it to the network. Modification of each message may have a different effect on the system state and depends solely on the design of the longitudinal control system. A comprehensive survey on attacks on the fundamental security goals, i.e., confidentiality, integrity, and availability can be found in [32]. In the *spoofing attack*, the adversary imitates another vehicle in the network to inject falsified messages into the target vehicle or a specific vehicle preceding the target. Therefore, the physical presence of the attacker close to the target vehicle is necessarily not required. In a recent study [33], the use of onboard ADAS sensors is proposed for the detection of location spoofing attack in vehicular networks. A similar type of attack in a vehicular network can be *GPS spoofing/jamming attack* [34] in which an attacker transmits false location information by generating strong GPS signals from a satellite simulator. In addition, a thief can use integrated GPS/GSM jammer to restrain a vehicle's anti-theft system from reporting the vehicle's actual location [35]. In the *replay attack*, the adversary stores the message received by one of the network's nodes and tries to replay it later to attain evil goals [36]. The replayed message contains old information that can cause different hazards to both the vehicular network and its nodes. For example, consider the message replaying attack by a malicious vehicle that is attempting to jam traffic [37].

(b) *Authenticity Attacks:* Authenticity is another major challenge in vehicular networks which refers to protecting the vehicular network from inside and outside malicious vehicles (possessing falsified identity) by denying their access to the system [38]. There are two types of authenticity attacks; namely, Sybil attack and impersonation attacks [39]. In a *Sybil attack*, a malicious vehicle pretends many fake identities [40] and in an *impersonation attack*, the adversary exploits a legitimate vehicle to obtain network access and performs malicious activities. For example, a malicious vehicle can impersonate a few non-malicious vehicles to broadcast falsified messages [41]. This type of attack is also known as the masquerading attack.

To avoid application layer attacks, various cryptographic approaches can be effectively leveraged especially when an attacker is a malicious outsider [10]. For instance, digital signatures can be used to ensure messages' integrity and to protect them against unauthorized use [42]. In addition, digital signatures can potentially provide both data and entity level authentication. Moreover, to prevent replay attacks, a timestamp-based random number (nonce) can be embedded within messages. While the aforementioned methods are general, there are other unprecedented challenges related to vehicular networks implementation, deployment, and standardization. For example, protection against security threats becomes more challenging with the presence of a trusted compromised vehicle with a valid certificate. In such cases, data-driven anomaly detection methods can be used [43], [44]. A survey on anomaly detection for enhancing the security of connected vehicles is presented in [45].

2) Network Layer Attacks: Network layer attacks are different from the application layer attacks in a way that they can be launched in a distributed manner. One prominent example of such attacks on vehicular systems is the use of vehicular botnets to attempt a denial of service (DoS) or distributed denial of service (DDoS) attack. The potential of vehicular network-based botnet attack for autonomous vehicles is presented in [46]. The study demonstrates that such an attack can cause severe physical congestion on hot spot road segments resulting in an increased trip duration of vehicles in the target area. Another way to realize the DoS attack is to use *network jamming* that causes disruption in the communications network over a small or large geographic area. As discussed earlier, current configurations of vehicular networks are based on the IEEE 802.11p standard that uses single control channel (CCH) with multiple service channels (SCH) and can be attacked by attempting single channel or multi-channel jamming by swiping between all channels. Various conventional techniques can be adopted to mitigate network layer attacks such as frequency hopping, channel, and technology switching, etc. *Coalition or platooning attack* is a similar type of attack in which a group of compromised vehicles can cooperate to perform malicious activities such as blocking or interrupting communications between legitimate vehicles.

3) System Level Attacks: The attacks on the vehicle's hardware and software are known as system level attacks and can be performed by either malicious insiders at the time of development or outsiders using unattended vehicular access. Such attacks are more serious in nature as they can cause damage even in the presence of the deployed state of the art security measures and secure end-to-end communications [47]. For instance, if the onboard hardware or software system of a vehicle is maliciously modified then the information exchange between the vehicle and communication systems will be inaccurate and with such a phenomenon the overall performance and security of the vehicular network will be compromised. In [48], authors investigated a non-invasive sensor spoofing attack on car's anti-lock braking system such that the braking system mistakenly reports a specific velocity.

4) Privacy Breaches: In vehicular networks, vehicles broadcast safety messages periodically that contain critical information such as vehicle identity, current location, velocity, acceleration, etc. The adversary can exploit such kind of information by attempting an *eavesdropping attack* which is a type of passive attack and is more difficult to be detected. Therefore, preserving the privacy of vehicles and drivers is of utmost importance. This allows the vehicles to communicate with each other without disclosing their identities, which is accomplished by masking their identities, e.g., using pseudonyms. In vehicular networks, knowing the origin of the message is crucial for authentication purposes, therefore, vehicles should be equipped with privacy-preserving authentication mechanism ensuring that the communication among

vehicles (V2V) and with infrastructure (V2I) is confidential. However, inter-vehicular communication can be eavesdropped by anyone within the radio range, e.g., a malicious vehicle can collect and misuse confidential information. Similarly, an attacker can construct location profiles of vehicles by establishing a connection with the RSU. Therefore, the effectiveness of pseudonymous or even complete anonymous schemes in vehicular networks remains vulnerable to privacy breaches [49].

5) Sensors Attacks: Although sensors of autonomous vehicles are by design resilient to environmental noises such as acoustical interference from nearby objects and vehicles, etc. However, current sensors cannot resist intentional noise and it can be injected to realize various attacks such as jamming and spoofing.

6) Attacks on Perception System: The perception system of self-driving vehicles is developed using various computer vision techniques including modern ML/DL-based methods for identifying objects, e.g., pedestrians, traffics signs, and symbols, etc. The perception system of self-driving vehicle is highly vulnerable to the physical world and adversarial attacks. For example, suppose we're learning a controller $f(x)$ to predict the steering angle in an autonomous car as a function of the vision-based input (captured into a feature vector x). The adversary may introduce small manipulations (i.e., x is modified into x') such that the predicted steering angle $f(x')$ is maximally distant from the optimal angle y.

7) Intrusion Detection: The detection of malicious activities is one of the major challenges of VANETs. Intrusion detection systems enable the identification of various types of attacks being performed on the system, e.g., sink- and black-hole attacks, etc. Without such a system, communication in vehicular networks is highly vulnerable to numerous attacks such as selective forwarding rushing, and Sybil attacks, etc. To detect the selective forwarding attack, a trust system based method utilizing local and global detection of attacks among inter-nodes mutual monitoring and detection of abnormal driving patterns is presented in [50]. Alheeti *et al.* proposed a system for intelligent intrusion detection of gray holes and rushing attack [51].

8) Certificate Revocation: The security mechanism of vehicular networks is based on trusted certification authority (CA) that manages the identities and credentials of the vehicles by issuing valid certificates to them. The vehicles are essentially unable to operate in the system without a valid certificate and validity of certificate must be revoked after a certain amount of time. The revocation process is a challenging task administratively due to challenges such as the identification of nodes with illegitimate behavior and the need to change the registered domain. Moreover, it is necessary to restrain malicious nodes by revoking their certificates to prevent them from attacking the system. To tackle this problem, three different certificate revocation protocols have been proposed in [52].

E. Non-Security-Related Challenges in Deploying CAVs

The phenomenon of connected vehicles is realized using a technology named vehicular networks which have various challenges that need to be addressed for their efficient deployment in the longer term that are described below.

1) High Mobility of Nodes: The large scale mobility of vehicles in vehicular networks result in a highly dynamic topology; thus, raising several challenges for the communication networks [53]. In addition, the dynamic nature of traffic can lead to a partitioned network having isolated clusters of nodes [54]. As the connections between the vehicles and nearby RSUs are short-lived, the wireless channel coherence time is short. This makes accurate real-time channel estimation more challenging at the receiver end. This necessitates the design of dynamic and robust resource management protocols that can efficiently utilize available resources while adapting to the vehicular density variations [55].

2) Heterogeneous and Stringent QoS Requirements: In vehicular networks, there are different modes of communications that can be broadly categorized into V2V and V2I communications. In V2V communications, vehicles exchange safety-critical information (e.g., information beacons, road and traffic conditions) among each other known as basic safety messages (BSM). This communication, which can be performed periodically or when triggered by some event, requires high reliability and is sensitive to delay [56].

In V2I communications, on the other hand, vehicles can communicate with nearby communications infrastructure to get support for route planning, traffic information, operational data, and to access entertainment services that requires more bandwidth and frequent access to the Internet, e.g., for downloading high-quality maps and accessing infotainment services, etc. Therefore, the heterogeneous and stringent QoS requirements of VANETs cannot be simultaneously met with traditional wireless design approaches.

3) Learning Dynamics of Vehicular Networks: As discussed above, vehicular networks exhibit high dynamicity; thus, to meet the real-time and stringent requirements of vehicular networks, historical data-driven predictive strategies can be adopted, e.g., traditional methods like hidden Markov models (HMM) and Bayesian methods [56]. In addition to using traditional ML methods, more sophisticated DL models can be used, for example, recurrent neural networks (RNN) and long short term memory (LSTM) have been shown beneficial for time series data and can be potentially used for modeling temporal dynamics of vehicular networks.

4) Network Congestion Control: Vehicular networks are geographically unbounded and can be developed for a city, several cities, and countries as well. The unbounded nature of vehicular networks leads to the challenge of network congestion [57]. As the traffic density is high in urban areas as compared to rural areas, particularly during rush hours, that can possibly lead to network congestion issues.

5) Time Constraints: The efficient application of vehicular networks requires hard real-time guarantees because it lays out the foundation for many other applications and services that require strict deadlines [58], for example, traffic flow prediction [59], traffic congestion control [60], and path planning [61]. Therefore, safety messages should be broadcasted in acceptable time either by vehicles or RSUs.

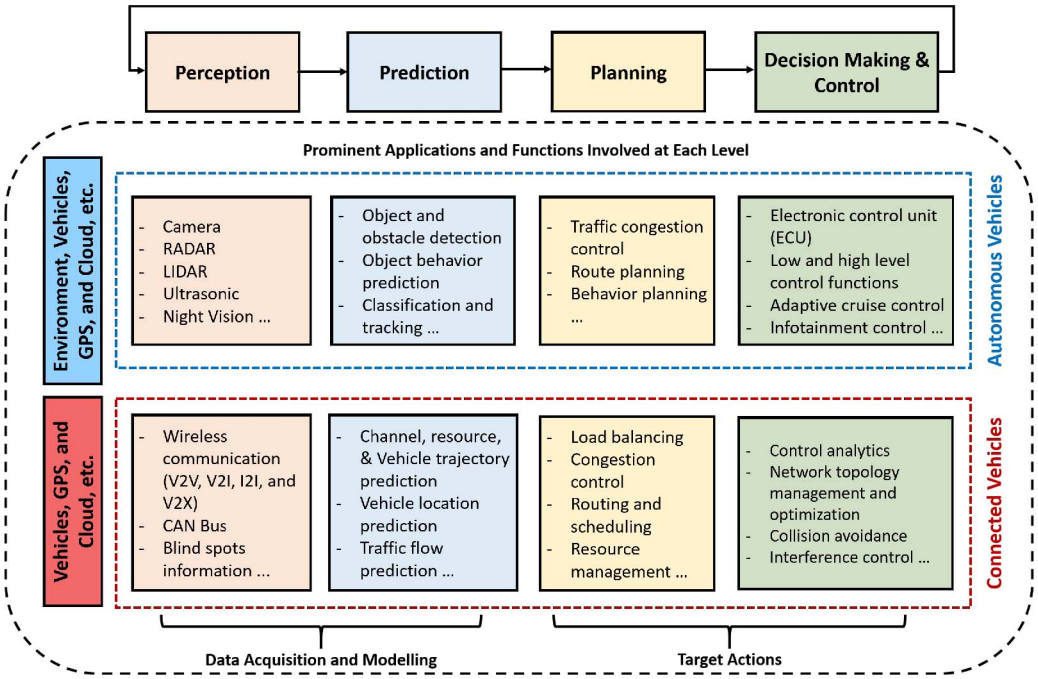

Fig. 8. The **machine learning (ML) pipeline** of CAVs comprising of four major modules: (1) perception; (2) prediction; (3) planning; and (4) control.

TABLE III
OVERVIEW OF MACHINE LEARNING (ML)-BASED RESEARCH ON DIFFERENT VEHICULAR NETWORK'S APPLICATIONS

Authors	Application	Methodology
Yao et al. [62]	Location prediction based scheduling and routing	Hidden Markov models
Xue et al. [63]		Variable-order Markov models
Zeng et al. [64]		Recursive least squares
Karami et al. [65]	Network congestion control	Feed forward neural network
Taherkhani et al. [66]		k-means clustering
Li et al. [67]	Load balancing	Reinforcement learning
Taylor et al. [68]	Network security	LSTM
Zheng et al. [69]	Virtual resource allocation	Reinforcement learning
Atallah et al. [70], [71]	Resource management	Reinforcement learning
Ye et al. [57]	Distributed resource management	Reinforcement learning
Kim et al. [72]	Vehicle trajectory prediction	Reinforcement learning

III. THE ML PIPELINE IN CAVS

The driving task elements of self-driving vehicles that can benefit from ML can be broadly categorized into the following four major components (as shown in Figure 8).

1) **Perception**: assists in perceiving the nearby environment and recognizing objects;
2) **Prediction**: predicting the actions of perceived objects, i.e., how environmental actors such as vehicles and pedestrians will move;
3) **Planning**: route planning of vehicle, i.e., how to reach from point A to B;
4) **Decision Making & Control**: making decisions relating to vehicle movement, i.e., how to make the longitudinal and lateral decisions to control and steer the vehicle.

These components are combined to develop a feedback system for enabling the phenomenon of self-driving without any human intervention. This ML pipeline can then facilitate autonomous real-time decisions by leveraging insights from the diverse types of data (e.g., vehicles' behavioral patterns, network topology, vehicles' locations, and kinetics information, etc.) that can be in easily gathered by CAVs.

In the remainder of this section, we will discuss some of the most prominent applications of ML-based methods for performing these tasks (a summary is presented in Table III).

A. Applications of ML for the Perception Task in CAVs

Different ML techniques, particularly, DL models have widely been used for developing the perception system of autonomous vehicles [73]. In addition to using video cameras as major visionary sensors, these vehicles also use other sensors for detection of different events in the car's surroundings, e.g., RADAR and LIDAR. The surrounding environment of the autonomous vehicles is perceived in two stages [74]. In the first stage, the whole road is scanned for the detection of changes in the driving conditions such as traffic signs and lights, pedestrian crossing, and other obstacles, etc. In the second stage, knowledge about the other vehicles is acquired. In [75], a CNN model is trained for developing direct perception representation of autonomous vehicles.

B. Applications of ML for the Prediction Task in CAVs

In CAVs, accurate and timely prediction of different events encountered in driving scenes is another important task which is mainly accomplished using different ML and DL algorithms. For instance, autonomous vehicles uses DL models for the detection and localization of obstacles [76], different objects (e.g., vehicles, pedestrians, and bikes, etc.) [77] and their behavior (e.g., tracking pedestrians along the way [78]) and traffic signs [79] and traffic lights recognition [80]. Another prediction tasks in CAVs that involve the application of ML/DL methods are vehicle trajectory and location prediction [81], efficient and intelligent wireless communication [82], and traffic flow prediction and modeling [83]. Moreover, ML schemes have also been used for the prediction of uncertainties in autonomous driving conditions [84].

C. Applications of ML for the Planning Task in CAVs

CAVs are equipped with onboard data processing compatibilities and they intelligently process the data collected from heterogeneous sensors for efficient route planning and for other optimized operations using different ML and DL techniques. The key goal of route planning is to reach the destination in a small amount of time while avoiding traffic congestion, potholes, and other vehicles by navigating through GPS and consuming less fuel as possible. In the literature, motion planning of autonomous vehicles is studied in three dimensions: (1) finding a path for reaching destination point; (2) searching for the fastest manoeuvre; and (3) determining the most feasible trajectory [85]. Moreover, to avoid collisions between vehicles in CAVs, predicting the trajectories of other vehicles is a crucial task for the planning trajectory of an autonomous vehicle [86]. For instance, Li presented a hybrid approach to model uncertainty in vehicle trajectory prediction for CAVs application using deep learning and kernel density estimation [87].

D. Applications of ML for the Decision Making and Control

In recent years, DL based algorithms have been extensively used for control of autonomous vehicles that are refined through millions of kilometers of test drives. For instance, Bojarski *et al.* presented a CNN based end-to-end learning framework for self-driving cars [88]. The model was able to drive the car on local roads with or without markings and on highways with small training data. In a similar study, CNN is trained for end-to-end learning of lane keeping for autonomous cars [89]. Recently, researchers have now started working on utilizing deep reinforcement learning (RL) for performing actions and decision making in driving conditions [90]. Bouton *et al.* proposed a generic approach to enforce probabilistic guarantees on RL learning for which they derived an exploration strategy that restricts the RL agent to choose among only those actions that satisfy a desired probabilistic specification criteria prior to training [91]. Moreover, human-like speed control of autonomous vehicles using deep RL with double Q-learning is presented in [92] that uses scenes generated by naturalistic driving data for learning. In [93], authors presented an integrated framework that uses a deep RL based approach for dynamic orchestration of networking, caching, and computing resources for connected vehicles.

In addition, ML-based methods have been used for many other applications in CAVs. For example, *adaptive traffic flow* in which smart infrastructure integrates V2V signals from the moving cars to optimize speed limits, traffic-light timing, and the number of lanes in each direction on the basis of the actual traffic load. The traffic flow can be further improved in CAVs by using *cooperative adaptive cruise control* technology [94]. Also, vehicles can take advantage of cruise control and save fuel by following one another in the form of vehicles platoons. Moreover, DL based methods have been proposed for intrusion detection for in-vehicle security of CAN bus [95]. The overview of intelligent and connected vehicles, current and future perspectives are presented in [96].

Autonomous vehicles are evolving through four stages of development. The first stage includes passive warning and convenience systems such as front and backward facing cameras, cross-traffic warning mechanism, radar for blind spot detection, etc. These warning systems use different computer vision and machine learning techniques to perceive the surrounding views of the vehicle on the road and to recognize traffic signs, static, and moving objects. In the second stage, these systems are used to assist the active control system of the vehicle while parking, braking, and to prevent backing over unseen objects. In the third stage, the vehicle is equipped with some semi-autonomous operations—as the vehicle may behave unexpectedly and the on seat driver should be able to resume control. In the final stage, the vehicle is designed to perform fully autonomous operations.

CAVs together formulate the settings of the self-driving vehicular network and there is a strong synergy between them [1]. In addition, autonomous vehicles are an important component of future vehicular networks that are equipped with complex sensory equipment. The autonomous vehicular networks are predictive and adaptive to their environments and are designed with two fundamental goals, i.e., autonomy and interactivity. The first goal enables the network to monitor, plan, and control itself and the later ensures that the infrastructure is transparent and friendly to interact with.

The deployment of ML in CAVs entails the following stages:

(a) *Data Collection:* Input data is collected using sensors or from other digital repositories. In autonomous vehicles, input data is collected using a complex sensory network, e.g., cameras, RADAR, GPS, etc. (see Figure 6); in a connected vehicular ecosystem, there is also inter-vehicle information communication.

(b) *(Pre-)Processing:* The heterogeneous data (video imagery, network, and traffic information, etc.) collected by the sensors is then digitally processed and appropriate features (e.g., traffic signs information and traffic flow information, etc.) are extracted.

(c) *Model Training:* Using the extracted features from the input data, a ML model is trained to recognize and distinguish between different objects events encountered in the driving environment, e.g., recognizing moving objects like pedestrian, vehicles, and cyclists, etc. and

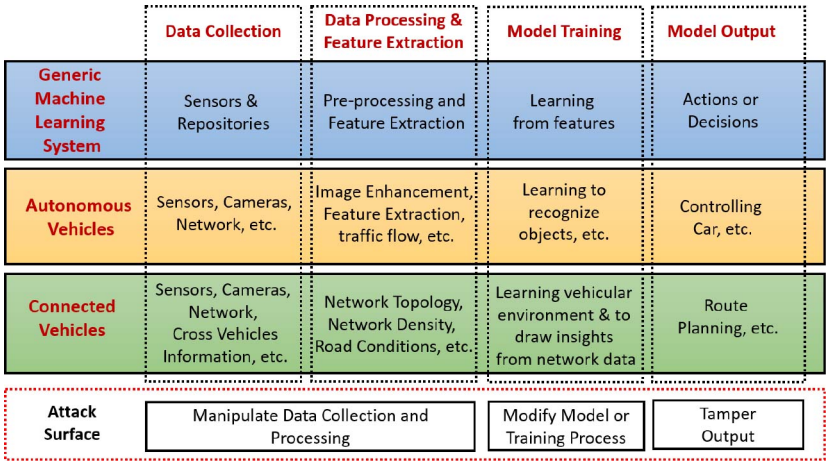

Fig. 9. The illustration of the generalization of attack surfaces in ML systems: generic model (top), autonomous vehicles model (middle), and connected vehicles model (bottom).

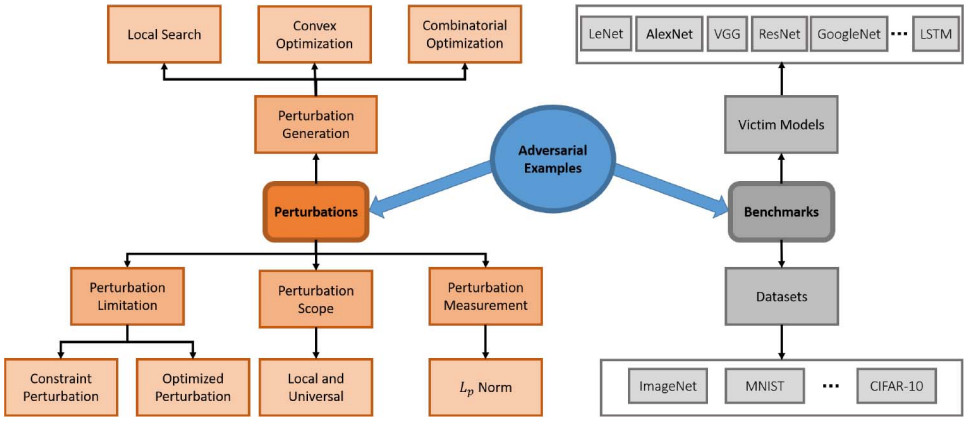

Fig. 10. The taxonomy of adversarial examples, perturbation methods, and benchmarks (datasets and models).

distinguishing between traffic signs, i.e., stop or speed limit sign, etc.

(d) *Decision or Action:* A decision or an action (e.g., stopping the car at the stop sign and predicting traffic flow based on the knowledge acquired by the vehicular network) is performed according to the learned knowledge and underlying system.

We present an illustration of the generalization of attack surfaces in ML systems from generic models to the more specific cases of autonomous and connected vehicles in Figure 9. As we shall discuss later in the paper, each of these stages is vulnerable to adversarial intrusion since an adversary can try to manipulate the data collection and processing system, tamper the model, or its outputs.

IV. Adversarial ML Attacks and the Adversarial ML Threat for CAVs

A comprehensive overview of adversarial ML in the context of CAVs is presented in this section.

A. Adversarial Examples

Formally, adversarial examples are defined as inputs to a deployed ML/DL model created by an attacker by adding an imperceptible perturbation in the actual input to compromise the integrity of the ML/DL model. An adversarial sample x^* is created by adding a small carefully crafted perturbation δ to the correctly classified sample x. The perturbation δ is calculated by approximating the optimization problem given in Eq. (1) iteratively until the crafted adversarial example gets classified by ML classifier $f(.)$ in targeted class t. A taxonomy of adversarial examples, perturbation methods, and benchmarks is presented in Figure 10.

$$x^* = x + \arg\min_{\delta}\{\|\delta\| : f(x+\delta) = t\}. \tag{1}$$

1) Adversarial Attacks: An adversarial attack affecting the training phase of the learning process is termed as poisoning attack where an attacker compromises the learning process of the ML/DL scheme by manipulating the training data [97], whereas the adversarial attack on the inference phase of the learning process is termed as evasion attack where an attacker manipulates the test data or real-time inputs to the deployed model for producing a false result [98]. Usually, examples used for fooling the ML/DL schemes at inference time are called adversarial examples.

2) Adversarial Perturbations: The adversarial perturbation crafting is divided in three major categories; namely, *local*

search, combinatorial optimization, and *convex relaxation*. This division is based on solving the objective function given in Eq. (1). Local search is the most common method of generating adversarial perturbations where the adversarial examples are generated by solving the objective function provided in Eq. (1) to obtain a lower bound on the adversarial perturbation by using gradient-based methods. A prime example of local search adversarial example crafting is the *fast gradient sign method* (FGSM) where an adversarial example is created by taking a step in the direction of the gradient [99]. In another study, the authors demonstrated that adversarial images are very easy to be constructed using evolutionary algorithms or gradient ascent [100]. Combinatorial optimization is also a method for creating adversarial examples where we find the exact solution of the optimization problem provided in Eq. (1), a major shortcoming of this method is the increase in the computational complexity with the increase of the number of examples in the dataset. Recently, Khalil *et al.* [101] launched a successful adversarial attack based on combinatorial and integer programming on binarized neural networks but the performance of the proposed attack reduces as the size and dimensions of data increase. Recently, convex relaxation is also used to generate [102] and defend [103] against adversarial examples where the upper bound on the objective function provided in Eq. (1) is calculated.

3) Different Aspects of Perturbations: The adversarial examples are designed to look like the original ones and imperceptible to humans. In this regard, the addition of small perturbations is of utmost importance. Whereas, the literature suggests that even one-pixel perturbation is often sufficient to fool the deep model trained for classification task [104]. Here we analyze different aspects of adversarial perturbations.

(a) *Perturbation Scope:* Adversarial perturbations are generated from two aspects: (1) perturbations for each legitimate input and (2) universal perturbations for the complete datasets, i.e., for each original cleaned sample. To date, most of the studies considered the first scope of adversarial perturbations.

(b) *Perturbation Limitation:* Similarly, there are two types of limitations, optimizing the system at a low perturbations scale and optimizing the system at a low perturbations scale with constrained optimization.

(c) The magnitude of the perturbations is mainly measured using three norms L_2, L_∞, and L_0 norm. In L_2-norm-based attacks, the attacker aims to minimize the squared error between the original and adversarial example. L_2-norm measures the Euclidean distance between the adversarial example and the original sample and results in a very small amount of noise added to the adversarial sample. L_∞ attacks are perhaps the simplest type of attacks which aim to limit or minimize the extent to which the maximum change for all pixels in adversarial examples is achieved. Also, this constraint forces to only make very small changes to each pixel. L_0-norm-based attacks work by minimizing the number of perturbed pixels in an image and force the modifications only to very few pixels.

To ensure tightly constrained action space available to an adversary, imperceptibility of perturbations is important to develop an attack. Considering the important constraints: (1) what constraints are placed on the attacker's "starting point"? and (2) where did this initial example come from? Gilmer *et al.* identified four salient features (described below) of adversarial perturbations [105].

(a) *Indistinguishable Perturbation:* The attacker does not have to select a starting point but it is given a draw from the data distribution and introduces such perturbation in the input sample that is indistinguishable by a human.

(b) *Content-Preserving Perturbation:* The attacker does not have to select a starting point but it is given a draw from the data distribution and creates such perturbation as long as the original content of the sample is preserved.

(c) *Non-suspicious Input:* The attacker can generate any type of desired perturbed input sample as long as it remains undetectable to a human.

(d) *Content-Constrained Input:* The attacker can generate any type of desired perturbed input sample as long as it maintains some content payload, i.e., it must be a picture of dog but not necessarily a particular dog. This includes payload-constrained input, where human perception might not be important. Rather, the intended function of the input example remains intact.

(e) *Unconstrained Input:* There is no constraint on the input and an attacker can produce any type of input example to get the desired output or behavior from the system.

4) Adversarial ML Benchmarks: In this section, we describe the benchmarks datasets and victim ML models used for evaluating adversarial examples. Researchers mostly adopt an inconsistent approach and report the performance of the attacks on diverse datasets and victim models. The widely used benchmark datasets and victim models are described below.

- *Datasets*: MNIST [106], CIFAR-10 [107], and ImageNet [108] are the widely used datasets in adversarial ML research and are also regarded as the standard deep learning datasets.
- *Victim Models*: The widely used victim ML/DL models for evaluating adversarial examples are LeNet [106], AlexNet [109], VGG [110], GoogLeNet [111], CaffeNet [112], and ResNet [113].

B. Threat Models for Adversarial ML Attacks on CAVs

Threat modeling is the procedure of answering a few common and straight forward questions related to the system being developed or deployed from a hypothetical attacker's point of view. Threat modeling is a fundamental component of security analysis. It requires that some fundamental questions related to the threat are addressed [114]. In particular, a threat model should identify:

- the *system principals:* what is the system and who are the stakeholders?
- the *system goals:* what does the system intend to do?
- the *system adversities:* what potential bad things can happen due to adverse situations or motivated adversaries?
- the *system invariants:* what must be always true about the system even if bad things happen?

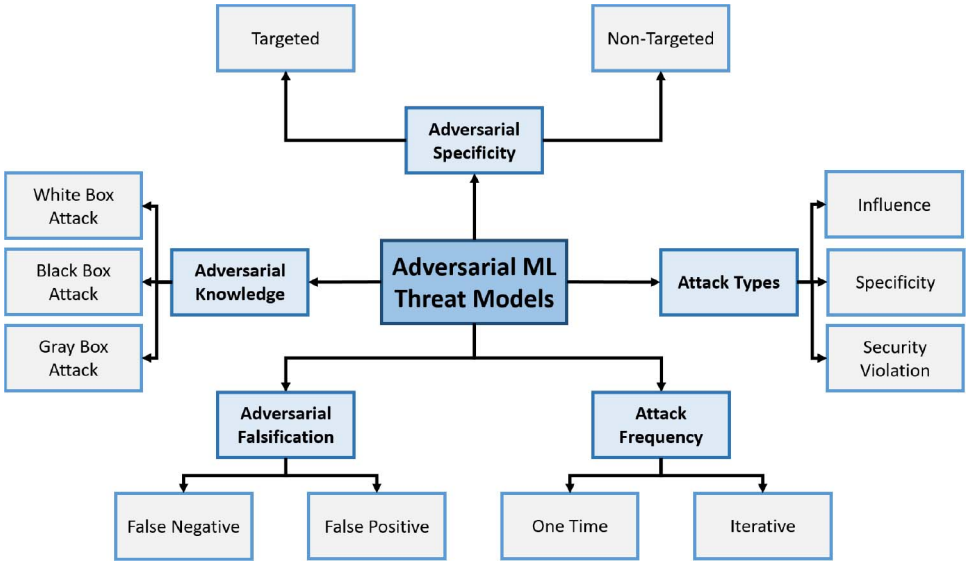

Fig. 11. The taxonomy of various types of threat models used in literature to design adversarial ML attacks. This figure also provides the information needed by a defender to ensure the robustness of ML-based autonomous system.

The key goal of threat modeling is to optimize the security of the system by determining security goals, identifying potential threats, and vulnerabilities, and to develop countermeasures for preventing or mitigating their associated effects on the system. Answering these questions requires careful logical thoughts and significant expertise and time.

As the focus of this paper is on highlighting the potential vulnerabilities of using ML techniques in CAVs, the scope of our study is restricted to the adversarial ML threat in CAVs. In the remainder of this section, we discuss the various facets of the adversarial ML threats in CAVs (a taxonomy aggregating these issues is illustrated in Figure 11).

1) Adversarial Attack Type: In the literature, the attacks on learning systems are generally categorized into three dimensions [115]:

(a) *Influence:* It includes causative (trying to get control over training data) and exploratory (exploiting misclassifications of the model without affecting the training process) attacks.
(b) *Specificity:* It involves targeted and indiscriminate attacks on a specific instance.
(c) *Security Violation:* It is concerned with the integrity of assets and availability of the service attack.

The first dimension describes the capabilities of the adversary and whether the attacker has the ability to affect the learning by poisoning training data. Instead, the attacker exploits the model by sending new samples and observing their responses to get the intended behavior. The second axis indicates the specific intentions of the attacker, i.e., whether the attacker is interested in realizing a targeted attack on one particular sample or he aims to cause learned model t fail in an indiscriminate fashion. The third dimension detail the types of security violation an attacker can cause, e.g., the attacker may aim to bypass harmful messages to bypass through the filter as false negatives or realizing denial of service by causing benign samples misclassified as false positives.

2) Adversarial Knowledge: Based on the adversarial knowledge available to the adversaries, the adversarial ML attacks are divided into three types; namely, *white-box, gray-box*, and *black-box* attacks. *White-box* attacks assume complete knowledge about the underlying ML model including information about the optimization technique, the trained ML model, model architecture, activation function, hyper-parameters, layer weights, and training data. Gray-box attacks assume a partial knowledge about the targeted model whereas the black-box adversarial attack assumes the adversary has zero knowledge and no access to the underlying ML model and the training data. Black box attack refers to the real-world knowledge where there is not much information about the targeted ML/DL scheme is available. In such cases, the adversary acts as a normal user and tries to infer from the output of the ML system. Black-box adversarial attacks make use of transferability property of adversarial examples where it is assumed that adversarial examples created for one ML/DL model will affect other models trained on datasets with a similar distribution to that of the original model [116].

3) Adversarial Capabilities: Adversarial capabilities are important to be identified in security practice. As they define the strength of the adversaries to compromise the security of the system. In general, an adversary can be stronger or weaker based on the knowledge and access to the system. Adversarial capabilities advocate how and what type of attacks an adversary can realize using what type of attack vector on which attack surface. The attacks can be launched at two main phases; namely, inference and training. Inference time attacks are exploratory attacks that do not modify the underlying model. Instead they influence it to produce incorrect outputs. Inference attacks vary with the availability of system knowledge. The training time attacks aim at tampering with the model itself or influence its learning process and involve two types of attack methods [117]. In the first type, adversarial

examples are injected in the training data and in the second type, training data is directly modified.

4) Adversarial Specificity: Another classification of the adversarial attacks is based on the specificity of the adversarial examples, where adversarial attacks are classified as *targeted* and *non-targeted* attacks. The attacks where adversarial perturbations are added to compromise the performance of a specific class in the data are known as the targeted adversarial attacks. Targeted adversarial attacks are launched by adversaries to create targeted misclassification (i.e., a specific road sign will be misclassified by the self-driving vehicle while the rest of the road sign classification system will function correctly) or source/target misclassification (i.e., a certain road traffic sign will be always classified in a pre-determined wrong class by the road sign classifier in a self-driving vehicle). Whereas adversarial perturbations created for deteriorating the performance of the model irrespective of any class of data are known as non-targeted adversarial attacks. Non-targeted attacks are launched by adversaries to reduce the classification confidence (i.e., a traffic sign will be detected with less accuracy which was previously detected with high accuracy) and misclassification (i.e., a road traffic sign will be classified in any other class than its original one).

5) Adversarial Falsification: The adversary can attempt two types of falsification attacks; namely, false positive attacks, and false negative attacks [15]. In the first attack, an adversary generates a negative sample which can be misclassified as a positive one. Let's assume such attack has been launched on the image classification system of an autonomous vehicle. A false positive will be an adversarial image predicted to be of a class with high confidence to whom it did not belong and is unrecognizable to humans. On the contrary, while attempting false negative attacks, the adversary generates a positive sample which can be misclassified as a negative one. In adversarial ML, this type of attack is referred to as an evasion attack.

6) Attack Frequency: The adversarial attacks can be single step or consist of an iterative optimization process. Compared to single step attacks, iterative adversarial attacks are stronger; however, they require frequent interactions for querying the ML system and subsequently require a large amount of time and computational resources for their efficient generation.

7) Adversarial Goals: The last component of the threat modeling is the articulation of the adversary's goals. The classical approach to model adversarial goals includes modeling of the adversary's desires to impact the confidentiality, integrity, and availability (known as the CIA model) and a fourth, yet important dimension is the privacy [117].

C. Review of Existing Adversarial ML Attacks

1) Adversarial ML Attacks on Conventional ML Schemes: A pioneering work on adversarial ML was performed by Dalvi *et al.* [125] in 2004 where they proposed a minimum distance evasion of the linear classifier and tested there proposed attack on spam classification system highlighting the threat of adversarial ML examples. A similar contribution was made by Lowd and Meek [126] in 2005 where they proposed adversarial classifier reverse engineering technique for constructing an adversarial attack on classification problems. In 2006, Barreno *et al.* [127] discussed the security of ML in adversarial environments and provided a taxonomy of attacks on ML schemes along with the potential defenses against them. In 2010, Huang *et al.* [128] provided the first consolidated review of adversarial ML where they discussed the limitations on the classifiers and adversaries in real-world settings. Biggio *et al.* [97] proposed poisoning attack on Support Vector Machines (SVM) to increase the test error in SVM, their attack successfully altered the test error of SVM with linear and non-linear kernels. The same authors also proposed an evasion attack where they used a gradient-based approach for evading PDF malware detectors [98] and tested their attack on SVM and simple neural networks.

2) Adversarial ML Attacks on DNNs: Adversarial ML attacks on DNNs were first observed by Szegedy *et al.* [9] where they demonstrated that DNNs can be fooled by minimally perturbing their input images at test time, the proposed attack was a gradient-based attack where minimum distance based adversarial examples were crafted to fool the image classifiers. Another gradient-based attack was proposed by Goodfellow *et al.* [99]. In this attack, they formulated adversarial ML as a min-max problem and adversarial examples were produced by calculating the lower bound on the adversarial perturbations. This method was termed as FGSM and is still considered a very effective algorithm for creating adversarial examples. Adversarial training was also introduced in the same paper as a defensive mechanism against adversarial examples. Kurakin *et al.* [118] highlighted the fragility of ML/DL schemes in real-world settings using images taken from a cell phone camera for adversarial example generation. The adversarial samples were created by using the basic iterative method (BIM) an extended version of FGSM. The resultant adversarial examples were able to fool the state-of-art image classifier. In [129], authors demonstrated that only rotation and translation are sufficient for fooling state-of-the-art deep learning based image classification models, i.e., convolutional neural networks(CNNs). In a similar study [130], ten state-of-the-art DNNs were shown to be fragile to the basic geometric transformation, e.g., translation, rotation, and blurring. Liu *et al.* presented a trojaning attack on neural networks that works by modifying the neurons of the trained model instead of affecting the training process [131]. Authors used trojan as a backdoor to control the trojaned ML model as desired and tested it on an autonomous vehicle. The car misbehaves when a specific billboard (trojan trigger) is encountered by it on the roadside.

Papernot *et al.* [6] exploited the mapping between the input and output of DNNs to construct a white-box Jacobian saliency-based adversarial attack (JSMA) scheme to fool the DNN classifiers. The same authors also proposed another defense against adversarial perturbations by using defensive distillation. Defensive distillation is a training method in which a model is trained to predict the classification probabilities of another model which was trained on the baseline standard to give more importance to accuracy. Papernot *et al.* [132] also proposed a black-box adversarial ML attack where they exploited the transferability property of adversarial examples

TABLE IV
SUMMARY OF THE STATE-OF-THE-ART ATTACKS

Year	Authors	Method	Adversarial Knowledge	Adversarial Specificity	Perturbation Scope	Perturbation Norm	Attack Learning
2014	Szegedy et al. [9]	L-BFGS	White box	Targeted	Image specific	L_∞	One Shot
2015	Goodfellow et al. [99]	FGSM	White box	Targeted	Image specific	L_∞	One shot
2016	Kurakin et al. [118]	BIM & ILCM	White box	Non targeted	Image specific	L_∞	Iterative
2016	Papernot et al. [6]	JSMA	White box	Targeted	Image specific	L_0	Iterative
2016	Moosavi et al. [119]	DeepFool	White box	Non targeted	Image specific	L_2,L_∞	Iterative
2017	Carlini et al. [120]	C&W attacks	White box	Targeted	Image specific	L_0,L_2,L_∞	Iterative
2017	Moosavi et al. [121]	Uni. perturbations	White box	Non targeted	Universal	L_2,L_∞	Iterative
2017	Sarkar et al. [122]	UPSET	Black box	Targeted	Universal	L_∞	Iterative
2017	Sarkar et al. [122]	ANGRI	Black box	Targeted	Image specific	L_∞	Iterative
2017	Cisse et al. [123]	Houdini	Black box	Targeted	Image specific	L_2,L_∞	Iterative
2018	Baluja et al. [124]	ATNs	White box	Targeted	Image specific	L_∞	Iterative
2019	Su et al. [104]	One-pixel	Black box	Non Targeted	Image specific	L_0	Iterative

to fool the ML/DL classifiers. This black-box adversarial attack was based on the substitute model training which not only fools the ML/DL classifiers but also breaks the distillation defensive mechanism. Carlini and Wagner [120] proposed a suite of three adversarial attacks termed as C&W attacks on DNNs by exploiting three distinct distance measures L_1, L_2, and L_∞. These attacks have not only evaded the DNN classifiers but also evaded the defensive distillation successfully. This demonstrated that defensive distillation is not an appropriate method for building robustness. In another paper, Carlini and Wagner [133] presented that the proposed adversarial attacks in [120] have successfully evaded the ten well known defensive schemes against adversarial examples. Right now these attacks are also considered as state-of-art adversarial ML attacks. Furthermore, Carlini and Wagner successfully demonstrated an adversarial attack on speech recognition system by adding small noise in the audio signal that forces the underlying ML model to generate intended commands/text [134]. In [135], an adversarial patch affixed to an original image forces the deep model to mis-classify that image. Such universal targeted patches fool classifiers without requiring knowledge of the other items in the scene. Such patches can be created offline and then broadly shared. More details on adversarial ML attacks can be found in [12], [13], [15], [16], [136], [137]. A summary of different state-of-the-art adversarial perturbation generation methods is provided in Table IV.

D. Adversarial ML Attacks on CAVs

ML and DL act as core ingredients for performing many key tasks in self-driving vehicles. Beyond providing deeply embedded information for the decision making process within the vehicle's components, they also play an important role in V2I and V2V, and V2X communications. As described in earlier sections, ML/DL schemes are very vulnerable to small carefully crafted adversarial perturbations. Self-driving vehicles are also threatened by this security risk along with other traditional security risks. Adversarial ML has affected many application domains including imaging, text, networking, and audio as highlighted in Table V.

1) Autonomous Vehicles Accidents Due to Unintended Adversarial Conditions: The autonomous vehicles developed so far are not robust to unintended adversarial conditions and there have been few reported fatalities caused by the malfunctioning of DNN-based autonomous vehicles where adversarial examples were unintentionally created by the DNN operating the autonomous vehicle. In 2014, during Hyundai competition, an autonomous vehicle crashed because of a sensor failure due to shifting in the angle of the car and direction of the sun.[2] Another incident was reported in 2016 where a Tesla autopilot was not able to handle the image contrast which resulted in the death of the driver.[3] It was also reported that the Tesla autopilot unable to differentiate between the bright sky and a white truck which resulted in a horrible accident. A similar accident happened to Google self-driving car where the car was unable to estimate the relative speed which resulted in a collision with a bus.[4] In 2018 Uber self-driving car also faced an accident due to malfunctioning in the DNN-based system which resulted in a pedestrian fatality.[5] Table VI provides a detailed description of accidents caused by malfunctions in different components of self-driving vehicles.

TABLE V
DOMAINS AFFECTED BY ADVERSARIAL MACHINE LEARNING (ML) AND THEIR APPLICATIONS

Domain	Application	Papers
Imaging	Digit Recognition	[6], [99], [120], ...
	Object Detection	[120], [132], [102], ...
	Traffic Signs Recognition	[132], [8], [138], ...
	Semantic Segmentation	[139], [140], [141], ...
	Reinforcement Learning	[142], [143], [144], ...
	Generative Modeling	[145], [146], [147], ...
Text	Text Classification	[148], [149], [150], ...
	Sentiment Analysis	[151], [150]
	Reading Comprehension	[152], [153]
Networking	Intrusion Detection	[154], [155], [156], ...
	Anomaly Detection	[157], [158]
	Malware Classification	[159], [160], [161], ...
	Traffic Classification	[162], [163]
Audio	Speech Recognition	[134], [164], [123], ...

2) Physical World Attacks on Autonomous Vehicles: Aung *et al.* [173] used FGSM and JSMA schemes to generate adversarial traffic signs to successfully evade the DNN-based traffic sign detection schemes to highlight the problem of adversarial examples in autonomous driving. Sitawarin *et al.* [138] proposed a real-world adversarial ML attack by altering the traffic signs and logos with adversarial perturbations while keeping the visual perception of the traffic and logo signs. In another work, Sitawarin *et al.* [8]

[2]https://bit.ly/2SWLxUY
[3]https://cnnmon.ie/2VOB283
[4]https://bit.ly/1U0O6yx
[5]https://bit.ly/2SWmb9N

TABLE VI
ACCIDENTS CAUSED BY SELF-DRIVING VEHICLES DUE TO UNINTENDED ADVERSARIAL CONDITIONS

Year	Company	Cause of accident	Damages	System failure
2014	Hyundai	Weather (Rain Fall)	Car crashed	Camera object detection failure
2016	Google Waymo	Speed estimation failure	Car crashed in the bus	Dynamic object movement detection failure
2016	Tesla	Image classification and image contrast failure	Car crashed in the neighbouring truck and the driver was killed	Camera's detection and classification suite failure
2017	Uber	Overreaction to an unseen event (Nearby accident)	Car crashed	Lack of robustness in control system
2017	General Motors	Stuck in dilemma (Lane change decision reversal)	Car knocked over a motorcyclist	Coordination and state estimation failure
2018	Uber	Confusion in the software decision system and safety system failure	Killed a person on the road	Failure of planning and perception system

proposed a technique for generating out-of-distribution adversarial examples to perform an evasion attack on ML-based sign recognition system of autonomous vehicles. They also proposed a Lenticular printing attack where they exploited the camera height in autonomous vehicles to create an illusion of false traffic signs in the physical environment to fool the sign recognition system of autonomous vehicles.

Object detection is another integral part of the perception module of autonomous vehicles where state-of-the-art DNN-based schemes such as Mask R-CNN [174] and YOLO [175] are used for object detection. Zhang *et al.* [167] proposed a camouflage physical world adversarial attack by approximately imitating how a simulator applies camouflage to the vehicle and then minimized the approximated detection score by using local search for optimal camouflage. The proposed adversarial attack successfully fooled image-based object detection systems. Another physical world adversarial example generation scheme on object detection is performed by Song *et al.* [168] where the perturbed "STOP" sign remained hidden from the state-of-art object detectors like Mask R-CNN and YOLO. They produced adversarial perturbations by the robust physical perturbations (RP_2) [169] algorithm. Recently Zhou *et al.* [166] proposed DeepBillboard a systematic way for generating adversarial advertisement billboards to inject a malfunction in the steering angle of the autonomous vehicle. The proposed adversarial billboard misled the average steering angle by 26.44 degrees. Table VII provides a summary of state-of-the-art adversarial attacks on self-driving vehicles. In a recent study [176], imitation learning has been shown robust enough for autonomous vehicles to drive in a realistic environment. Authors proposed a model named ChauffeurNet that learns to drive the vehicle by imitating best and synthesizing worst.

V. TOWARDS DEVELOPING ADVERSARIALLY ROBUST ML SOLUTIONS

As discussed above, despite the outstanding performance of ML techniques in many settings, including human level accuracy at recognizing images. These techniques exhibit strict vulnerability to carefully crafted adversarial examples. In this section, we present an outline of approaches for developing adversarially robust ML solutions. We define the robustness as the ability of the ML model to restrain adversarial examples.

In the literature, defenses against adversarial attacks have been divided into two broad categories: (1) *reactive* detect adversarial observations (input) after deep models are trained; and (2) *proactive* make the deep model robust against adversarial examples before the attack.

Alternatively, these techniques can also be broadly divided into three categories: (1) modifying data; (2) adding auxiliary models; and (3) modifying models. The reader is referred to Figure 12 for a visual depiction of a taxonomy of robust ML solutions in which various techniques that fall in these categories are also listed. These categories are detailed next.

A. Modifying Data

The methods falling under this category mainly deal with modification of either the training data (e.g., adversarial retraining) and its features or test data (e.g., data preprocessing). Widely used approaches that utilize such methods are described below.

1) Adversarial (Re-)Training: The training with adversarial examples has been firstly proposed by Goodfellow *et al.* [99] and Huang *et al.* [177] as a defense strategy to make deep neural networks (DNNs) robust against adversarial attacks. They trained the model by augmenting adversarial examples in the training set. Furthermore, Goodfellow *et al.* showed that adversarial training could provide better regularization for DNNs. In [99], [177], the adversarial robustness of ML models was evaluated on the MNIST dataset having 10 classes while in [178], a comprehensive evaluation of adversarial training was performed on a considerably large dataset, i.e., ImageNet having 1000 classes. The authors used 50% of the dataset for adversarial training and this strategy increased the robustness of DNNs for single step adversarial attack (e.g., FGSM [99]). However, the strategy failed for iterative adversarial examples generation methods such as the basic iterative method (BIM) [118].

2) Input Reconstruction: The idea of input reconstruction is to clean the adversarial examples to transform them back to legitimate ones. Once the adversarial examples have been transformed, they will not affect the prediction of DNN models. For robustifying DNN, a technique named deep contractive autoencoder has been proposed in [179]. They trained a denoising autoencoder for cleaning adversarial perturbations.

3) Feature Squeezing: Xu *et al.* [180] leveraged the observation that input feature spaces are typically unnecessarily large and provide a vast room for an adversary to construct adversarial perturbations and thereby proposed feature squeezing as a defense strategy to adversarial examples. The available feature space to an adversary can be reduced using feature squeezing that combines samples having heterogeneous feature vectors in the original space into a single space. They perform feature squeezing at two levels: (1) reducing color bit depth; (2) spatial domain smoothing using both local and non-local method. Also, they evaluated eleven state-of-the-art adversarial perturbations generation methods on three different datasets, i.e., MNIST, CIFAR-10, and ImageNet. However, this defense strategy was found to be less effective in a later study [181].

TABLE VII
ADVERSARIAL ATTACKS ON SELF-DRIVING VEHICLES: SUMMARY OF STATE-OF-THE-ART

Attack Objective	Specific work	Problem formulation	Data	Threat model	Attack results
Perception system failure	DARTS [8]: Traffic signs manipulation	Generating adversarial examples for CNN-based traffic sign detection by performing out-of-distribution attacks along with Lenticular printing.	GTSRB GTSDB	1) Virtual and physical world attack 2) White and black-box mode	DARTS has successfully fooled the perception system of self-driving car.
Perception system failure	Rogue Signs [138]: Traffic signs and logos manipulation	End-to-end pipeline for adversarial example generation for CNN-based traffic signs and logo detection.	GTSRB	1) Virtual and physical world attack 2) White-box mode	Fooled the perception system of self-driving car with a success rate of 99.7%.
Object detection failure	ShapeShifter [165]: Adversarial attack on Faster R-CNN	Adversarial attack on bounding boxes of Faster R-CNN by using expectation over transformation techniques.	MS-COCO	1) Virtual and physical world attack 2) White-box mode	Caused malfunction in Faster R-CNN of self-driving car with 93% success.
Motion planning and perception system failure	DeepBillboard [166]: Adversarial attack through drive-by billboards.	Adversarial attack on steering angle of the self-driving car by using adversarial perturbations in drive-by billboards.	1) Udacity self-driving car challenge dataset 2) Dave testing dataset 3) Kitti dataset	1) Virtual and physical world attack 2) White and black-box mode	Caused a 23 degree malfunction in steering angle of self-driving car.
Object detection failure	CAMOU [167]: Adversarial camouflage to fool the object detector	Generating adversarial perturbation to prevent the self-driving car from Mask R-CNN-based object detector.	Unreal engine simulator	1) Virtual and physical world attack 2) White and black-box mode	Caused a 32.74% drop in the performance of Mask R-CNN.
Object detection failure	Song et al. [168]: Object disappearance and creation attack	Generating adversarial perturbation based stickers where object detection schemes like Yolo and R-CNN used in self-driving cars fails to recognize certain signs and logos. Furthermore object detection schemes start detecting things that are not present in the frame.	Video of traffic signs	1) Virtual and physical world attack 2) White-box mode	The object detection schemes fails to recognize traffic signs nearly in 86% of the frames in the video.
Perception system failure	Eykholt et al. [169]: Robust physical perturbations against visual classification under different physical environment.	Robust physical perturbations are created to fool LISA-CNN and GTSRB-CNN based traffic sign classification schemes.	1) LISA 2) GTSRB	1) Virtual and physical world attack 2) White and black-box mode	In few cases caused 100% performance drop in visual classification of traffic signs
Perception and controller system failure	Tuncali et al. [170]: Simulation based adversarial test generation for self-driving cars.	Testing and verifying self-driving car's perception and controller system against adversarial examples.	Simulated data from proposed simulator	1) Virtual environment 2) White and black-box mode	Designed system was able to detect critical cases in autonomous car's perception and control.
Controller system failure	Yaghoubi et al. [171]: Finding gray-box adversarial examples for closed loop autonomous cars control system.	Testing the controller and perception system of self-driving cars against gradient-based gray-box adversarial examples.	Simulated data	1) Virtual environment 2) Gray box mode	Gray-box adversarial examples have outperformed simulated Annealing optimization in a dummy control system problem.
End-to-end autonomous control failure	Boloor et al. [172]: Physical adversarial examples against E2E driving models.	Disrupting steering by using physical perturbation in the environment.	CARLA simulated data	1) Virtual environment 2) White-box mode	Physical adversarial perturbation has forced the self-driving car to crash.

4) Features Masking: In [182], authors proposed to add a masking layer before the softmax layer of the classifier that is mainly responsible for the classification task. The purpose of adding the masking layer was to mask the most sensitive features of input that are more prone to adversarial perturbations by forcing the corresponding weights of this layer to zero.

5) Developing Adversarially Robust Features: This method has been recently proposed as an effective approach to make DNNs resilient against adversarial attacks [183]. Authors leveraged the connections between the natural spectral geometrical property of the dataset and the metric of interest for developing adversarially robust features. They empirically demonstrated that the spectral approach can be effectively used to generate adversarially robust features that can be ultimately used to develop robust models.

6) Manifold Projection: In this method, input examples are projected on the manifold of learned data from another ML model, generally, the manifold is provided by a generative model. For instance, Song *et al.* [184] leveraged generative models to clean the adversarial perturbations from malicious images and then the cleaned images are given to the non-modified ML model. Furthermore, this paper ascertains that regardless of the attack type and targeted model, the adversarial examples lie in the low probability regions of the training data distribution. In a similar study [185], authors used generative adversarial networks (GANs) for cleaning adversarial perturbations. Similarly, Meng and Chen proposed a framework named MagNet that includes one or more detectors and a reformer network [186]. The detector network is used to classify normal and adversarial examples by learning the manifold of normal examples, whereas, the reformer network moves adversarial examples towards the learned manifold.

B. Modifying Model

The methods that fall in this category mainly modify the parameters/features learned by the trained model (e.g., defensive distillation), a few prominent such methods are described next.

1) Network Distillation: Papernot *et al.* [187] adopted network distillation as a procedure to defend against adversarial attacks and presented a defense method known as defensive distillation. The notion of distillation was originally proposed by Hinton *et al.* [188] as a mechanism for effectively transferring knowledge from a larger network to a smaller one. The defense method developed by Papernot *et al.* uses the probability distribution vector generated by the first model as an input to the original DNN model. This increases the resilience of the DNN model towards very small perturbations. However, Carlini and Wagner showed that the defensive distillation method does not work against their proposed attack [133].

2) Network Verification: Network verification aims to verify the properties of DNN, i.e., whether an input satisfies or violates certain property because it may restrain new unseen adversarial perturbations. For instance, a network verification method for robustifying DNN models using ReLU activation is presented in [178]. To verify the properties of the deep model, the authors used the satisfiability modulo theory (SMT) solver and showed that the network verification problem is

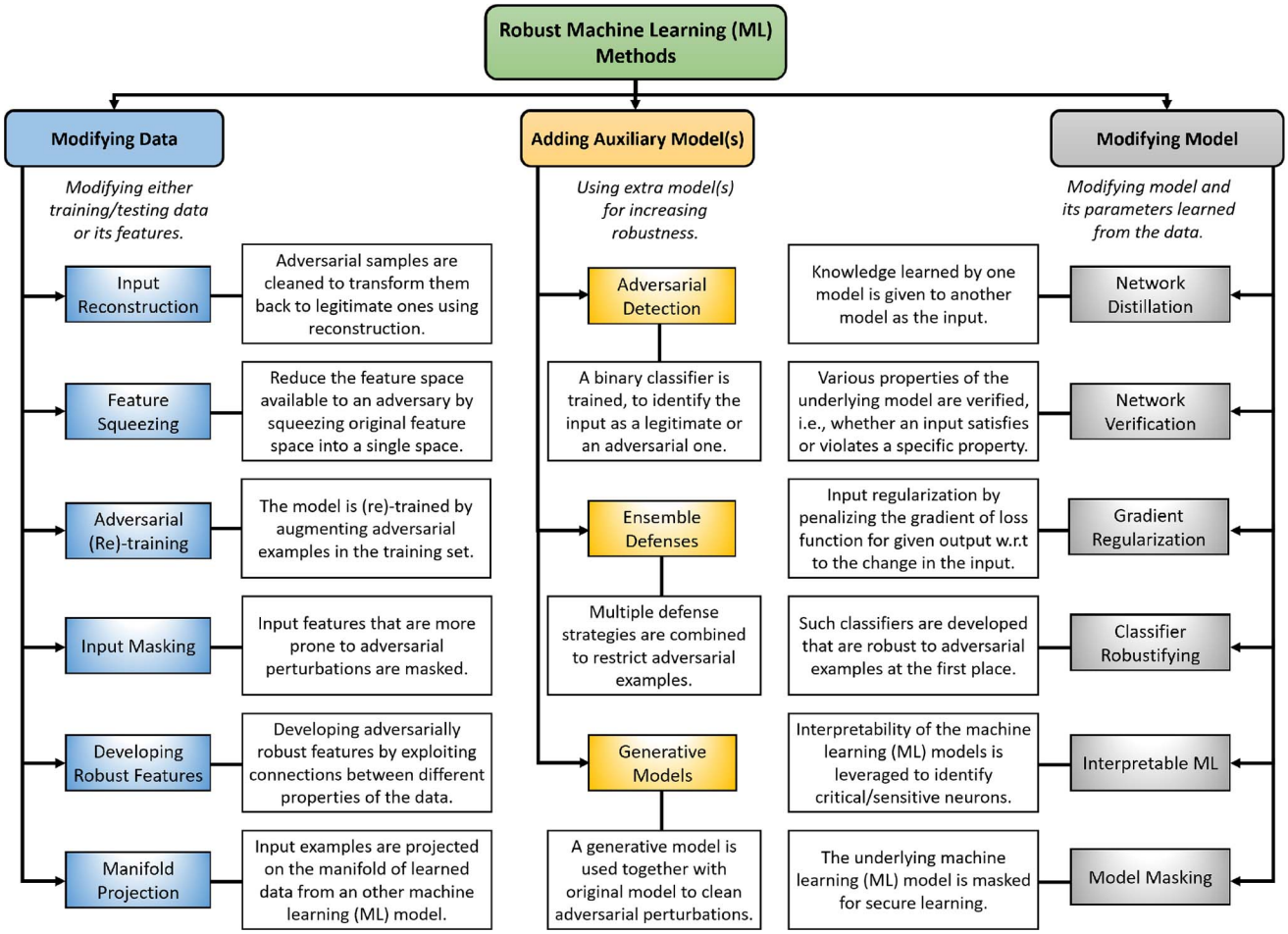

Fig. 12. Taxonomy of robust machine learning (ML) methods categorized into three classes: (1) Modifying Data (2) Adding Auxiliary Model(s) and (3) Modifying Models.

NP-complete. The assumption of using ReLU with certain modifications is addressed in [189].

3) Gradient Regularization: Ross and Doshi-Velez [190] proposed using input gradient regularization as a defense strategy against adversarial attacks. In the proposed approach, they used differentiable DNN models and penalized the variation that results in the output with a change in the input. As a result, adversarial examples with small perturbations were unlikely to modify the output of deep models but this increases the training complexity with a factor of two. The notion of penalizing the gradient of loss function of models with respect to the inputs for robustification has been already been investigated in [191].

4) Classifier Robustifying: In this method, classification models that are robust to adversarial attacks are designed from the ground up instead of detecting or transforming them. Bradshaw *et al.* [192] utilized the uncertainty around the adversarial examples and developed a hybrid model using Gaussian processes (GPs) with RBF kernels on top of DNNs and showed that their approach is robust against adversarial attacks. The latent variable in GPs is expressed using a Gaussian distribution and is parameterized by mean and covariance and encoded with RBF kernels. Schott *et al.* [193] proposed the first adversarially robust classifier for MNIST dataset, where robustness is achieved by using analysis by synthesis through learned class-conditional data distribution. This work highlights the lack of research that provides guaranteed robustness against adversarial attacks.

5) Explainable and Interpretable ML: In a recent study [194], an adversarial example detection approach is presented for a face recognition task that leverages the interpretability of DNN models. The key in this approach is the identification of critical neurons for an individual task that is performed by establishing a bi-directional correspondence inference between the neurons of a DNN model and its attributes. Then the activation values of these neurons are amplified to augment the reasoning part and the values of other neurons are decreased to conceal the uninterpretable part. Recently, Nicholas Carlini showed that this approach does not defend against untargeted adversarial perturbations generated using L_∞ norm with a bound of 0.01 [195].

6) Masking ML Model: In a recent study [196], authors formulated the problem of adversarial ML as learning and masking problem and presented a classifier masking method for secure learning. To mask the deep model, they introduced noise in the DNN's logit output that was able to defend against low distortion attacks.

C. Adding Auxiliary Model(s)

These methods aim to utilize additional ML models to enhance the robustness of the main model (e.g., using generative models for adversarial detection), such widely used methods are described as follows.

1) Adversarial Detection: In adversarial detection strategy, a binary classifier (detector) is trained, e.g., DNN to identify the input as a legitimate or an adversarial one [197], [198]. In [199], authors used a simple DNN-based binary adversarial detector as an auxiliary network to the main model. In a similar study [200], authors introduced an outlier class while training the DNN model, the model then detects the adversarial examples by classifying them as an outlier. This defense approach has been used in a number of studies in the literature.

2) Ensembling Defenses: As adversarial examples can be developed in a multi-facet fashion, therefore, multiple defense methods can be combined together (parallelly or sequentially) to defend against them [201]. PixelDefend [184] is a prime example of ensemble defense in which an adversarial detector and an "input reconstructor" are integrated to restrain adversarial examples. However, He *et al.* showed that an ensemble of weak defense strategies does not provide a strong defense to adversarial attacks [181]. Further, they demonstrated that adaptive adversarial examples transfer across several defense or detection proposals.

3) Using Generative ML Models: Goodfellow *et al.* [99] firstly coined the idea of using generative training to defend adversarial attacks, however, in the same study they argued that being generative is not sufficient and presented an alternative hypothesis of ensemble training that works by ensembling multiple instances of original DNN models. In [202], an approach named cowboy is presented to detect and defend against adversarial examples. They transformed adversarial samples back to data manifold by cleaning them using a GAN trained on the same dataset. Furthermore, authors empirically showed that adversarial examples lie outside the data manifold learned by the GAN, i.e., the discriminator of GAN consistently scores the adversarial perturbations lower than the real samples across multiple attacks and datasets. In another similar study [203], a GAN-based framework named Defense-GAN is trained for modeling the distribution of legitimate images. During inference time, Defense-GAN finds a similar output without adversarial perturbations that is then fed to the original classifier. Also, the authors of both of these studies claimed that their method is independent of the DNN model and attack type and that it can be used in existing settings. The summary of various state-of-the-art adversarial defense studies is presented in Table VIII.

D. Potential of Adversarial Defenses for CAVs

The adversarial defense methods described in the above sections are general, i.e., they are developed to make DNN resilient against adversarial attacks. However, these methods have a great potential to be used for the robust application of DL models in CAVs settings and can be used for robustifying various applications (as described in Section III) of ML/DL in CAVs ecosystem. For instance, defense methods aiming at robustifying the DNN classifier can be used for developing adversarially robust objection detection systems. Also in CAVs, the presence of adversaries who may want to harm the CAVs environment is more common, e.g., consider an adversary trying to get the control of the autonomous car in an attempt to force the vehicle to cause an accident, etc. and the above mentioned general adversarial defense methods can provide robustness to ML/DL applications in CAVs.

However, more research is needed to design, implement, and experiment with new adversarial defenses that are customized from the ground-up for CAVs. Such defenses should benefit from existing general adversarial defense techniques while taking CAVs safety, delay, and computational constraints into considerations. We believe that such defenses can hasten the deployment of ML models in practical CAVs settings.

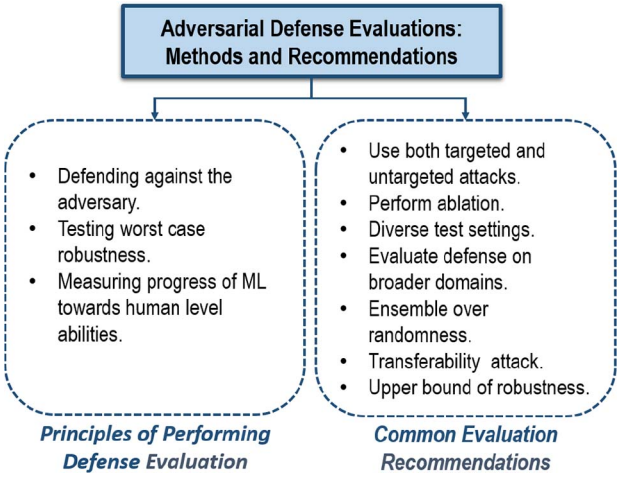

Fig. 13. The taxonomy of different adversarial defense evaluation methods and recommendations.

E. Adversarial Defense Evaluation: Methods and Recommendations

This section presents different potential methods for performing the evaluation of adversarial defenses along with an outline of common evaluation recommendations, as depicted in Figure 13.

1) Principles for Performing Defense Evaluations: In a recent study [212], Carlini *et al.* provided recommendations for evaluating adversarial defenses and thereby provided three common reasons to evaluate the performance of adversarial defenses. These recommendations are briefly described below.

a) Defending against the adversary: Defending against adversaries attempting adversarial attacks on the system is crucial as it is a matter of security concern. In real-world applications, if the ML-based systems are deployed without considering the security threats then the adversaries willing to harm the system will continue to practice attacking the system as long as there are incentives. The nature and sovereignty of attacks vary with adversarial capabilities and knowledge, etc. In this regard, proper and well-thought threat modeling (described in detail in an earlier section) is of paramount importance.

TABLE VIII
SUMMARY OF STATE-OF-THE-ART ADVERSARIAL DEFENSE APPROACHES

Reference	Type	Defense	Adv. Perturbation	Dataset	Threat Model	Original Accuracy	Adversarial Accuracy	Defense Success
Gu et al. [179]	Modifying Data	Adversarial examples cleaning using denoising autoencoders (DAEs).	Local perturbations, e.g., additive Gaussian noise.	MNIST	Not clearly articulated	99 (%)	100 (%)	99.1%
Xu et al. [180]		Reduced the feature space available to an adversary.	Evaluated different state of the art perturbation generation methods.	MNIST, CIFAR-10, and ImageNet	White Box	MNIST (99.43%), CIFAR-10 (94.84%), and ImageNet (68.36%)	Roughly achieved 100% for each model using different attack algorithms.	MNIST (62.7%), CIFAR-10 (77.27%), and ImageNet (68.11%)
Gao et al. [182]		Proposed DeepCloak that removes unnecessary features in the model.	Perturbations are generated using FGSM.	CIFAR-10	Not clearly articulated	93.72% (1% masking)	39.23% (1% masking)	10% increase in adversarial settings
Garg et al. [183]		Constructed adversarially robust features using spectral property of the dataset.	L_2 Perturbations	MNIST	Not clearly articulated	Not clearly articulated	Not clearly articulated	Provided empirical evidence for the effectiveness of the proposed defense.
Song et al. [184]		Proposed PixelDefend to clean adversarial examples by moving them back to the manifold of original training data.	Used five state of the art adversarial attacks.	Fashion MNIST and CIFAR-10	White Box	90%	Fashion MNIST (63%), CIFAR-10 (32%) for strongest defense.	Adversarial accuracy increased from 63% to 84% for Fashion MNIST and 32% to 70% for CIFAR-10.
Prakash et al. [204]		Used wavelet-based denoising method to clean natural and adversarial noise.	Generated perturbations using pixel deflection $L_2(\epsilon = 0.05)$.	ImageNet	White Box	98.9%	Not clearly articulated	81% accuracy
Xie et al. [205]		Proposed to use randomization at inference stage and used random resizing and random padding.	$L_\infty(\epsilon = 10/255)$	ImageNet	White Box and Black Box	99.2%	Not clearly articulated	86% accuracy
Guo et al. [206]		Investigated different image transformation methods defending adversarial attacks.	$L_2(\epsilon = 0.06)$	ImageNet	Gray Box and Black Box	75%	Not clearly articulated	70% accuracy
Goodfellow et al. [99]		Augmented adversarial examples into the training set.	Fast Gradient Sign (FGSM) method	MNIST	Not clearly articulated	Not clearly articulated	99.9%	With adversarial training, the error rate fell to 17.9%
Schott et al. [193]	Adding Auxiliary Model	Used generative modelling using variational autoencoder (VAE.)	Applied score-based, decision-based, transfer-based, and gradient-based attacks using $L_2(\epsilon = 1.5)$.	MNIST	Not clearly articulated	99%	Not clearly articulated	80%
Wong et al. [103]		Formulated a robust optimization problem using convex outer approximation for detection of adversarial examples.	FGSM and gradient descent based methods, $L_\infty(\epsilon = 0.1)$.	MNIST	Not clearly articulated	98.2% accuracy	Not clearly articulated	94.2% accuracy
Liao et al. [207]		Proposed a high-level representation guided denoiser (HGD) for defending adversarial attacks.	$L_\infty(\epsilon = 4/255)$	ImageNet	White Box and Black Box	75%	Not clearly articulated	75% accuracy
Ross et al. [190]	Modifying Model	Trained the model with input gradient regularization for defending adversarial attacks.	Evaluated three famous attacks, i.e., FGSM, TGSM, and JSMA.	MNIST, Street-View House Numbers (SVHN), and notMNIST	White Box and Black Box	Not clearly articulated	Not clearly articulated	MNIST (100%), SVHN ($\sim$90%), and notMNIST (100%)-approximately same for each type of attack.
Madry et al. [208]		Trained the model with optimized parameters using robust optimization.	$L_\infty(\epsilon = 8/255)$	CIFAR-10	White Box and Weaker Black Box	87%	Not clearly articulated	46% accuracy
Buckman et al. [209]		Proposed to use thermometer encoding for inputs.	$L\infty(\epsilon = 8/255)$	CIFAR-10	White Box	90%	Not clearly articulated	79% accuracy
Dhillon et al. [210]		Proposed stochastic activation pruning of the trained model for defense.	$L_\infty(\epsilon = 4/255)$	CIFAR-10	Not clearly articulated	83%	Not clearly articulated	51% accuracy
Croce et al. [211]		Proposed a regularization scheme for ReLU networks	Perturbations using L_2 and L_∞ methods.	MNIST, German Traffic Signs (GTS), Fashion MNIST, and CIFAR-10.	Not clearly articulated	98.81%	Not clearly articulated	96.4% accuracy (on first 1000 test points)

b) Testing worst-case robustness: In real-world settings, testing the worst-case robustness of ML models from the perspective of an adversary is crucial as real-world systems exhibit randomness that is hard to be predicted. Compared to the random testing approach, worst-case analysis can be a powerful tool to distinguish a system that fails one time in

a billion trials from a system that never fails. For instance, if a powerful adversary who is attempting to harm a system to get intentional misbehavior fails to do so, then it provides strong evidence that the system will not misbehave in case of previously unforeseen randomness.

c) Measuring progress of ML towards human level abilities: To advance ML techniques, it is important to understand why ML algorithms fail in a particular setting. In the literature, we see that the performance gap between ML methods and humans is considerably small on many complex tasks, e.g., natural image classification [109], mastering the game of Go using reinforcement learning [213], and human level accuracy in the medical domain [214], [215]. However, in case of evaluating adversarial robustness, the performance gap between humans and ML systems is very large. This is so true for the cases where ML models exhibit super-human accuracy, i.e., an adversarial attack can completely evade the prediction performance of the system. This leads to the belief that there exists a fundamental difference between the decision making process of humans and ML models. So, keeping this aspect in mind, adversarial robustness is the measure of ML progress that is orthogonal to performance.

2) Common Evaluation Recommendations: In this section, we provide a brief discussion on the common evaluation recommendations and we refer interested readers to the recent article of Carlini *et al.* [212] for a detailed and comprehensive description on evaluation recommendations and pitfalls for adversarial robustness. As authors promised to update this paper timely, therefore, we also refer interested readers to following URL[6] for an updated version of this paper. To avoid unintended consequences and pitfalls of evaluation methods, the following evaluation recommendations can be adopted.

a) Use both targeted and untargeted attacks: Adversarial robustness should be evaluated on both targeted and untargeted attacks. In any case, it is important to explicitly state which attack were considered while evaluating. Theoretically, an untargeted attack is considered to be strictly easier than a targeted attack but practically, performing an untargeted attack can give better results than targeting any of $N - 1$ classes. Many untargeted attacks mainly work by minimizing the prediction confidence of the correct label. Contrarily, targeted attacks work by maximizing the prediction confidence of some other class.

b) Perform ablation: Perform ablation analysis by removing a combination of defense components and verifying that the attack succeeds on a similar but undefended model. This is useful to develop a straight forward understanding of the goals of the evaluation and assess the effectiveness of combining multiple defense strategies for robustifying the model.

c) Diverse test settings: Perform the evaluation in diverse settings, i.e., test the robustness to random noise, validate broader threat models, and carefully evaluate the attack hyperparameters and select those that provide the best performance. It is also important to verify that the attack converges under selected hyperparameters. Also, investigate whether attack results are sensitive to a specific set of hyperparameters. In addition, experiment witg at least one hard label attack and one gradient free attack.

d) Evaluate defense on broader domains: For a defense to be truly effective, consider evaluating the proposed defense method on broader domains other than images. For instance, the majority of works on adversarial machine learning mainly investigate the imaging domain. State explicitly if the defense is only capable of defending adversarial perturbations in a specific domain (e.g., images).

e) Ensemble over randomness: It is important to create adversarial examples by ensembling over the randomness of those defenses that randomize aspects of DNN inference. The introduced randomness enforces stochasticity and standard attacks become hard to be realized. Verify that the attack remains successful when randomness is assigned a fixed value. Also, define the threat model and the availability of randomness knowledge to the adversary.

f) Transferability attack: Select a similar substitute model (to the defended model) and perform transferability of the attack. Because the adversarial examples are often transferable across different models, i.e., an adversarial sample constructed for one model often appears adversarial to another model with identical architecture [216]. This is true regardless of the fact that the other model is trained on completely different data distribution.

g) Upper bound of robustness: To provide upper bound on robustness, apply adaptive attacks, i.e., give access to a full defense. Apply the strongest attack for a given threat model and defense being evaluated. Also, verify that adaptive attacks perform better than others and evaluate their performance in multiple settings, e.g., the combination of transfer, random-noise, and black-box attacks. For instance, Ruan *et al.* evaluated the robustness of DNN and presented an approximate approach to provide lower and upper bounds on robustness for L_0 norm with provable guarantees [217].

F. Testing of ML Models and Autonomous Vehicles

1) Behavior Testing of Models: In a recent study, Sun *et al.* proposed four novel testing criteria for verifying structural features of DNN using MC/DC[7] coverage criteria [218]. They validated proposed methods by generating test cases guided by their proposed coverage criteria using both symbolic and gradient-based approach and showed that their method was able to capture undesired behaviors of DNN. Similarly, a set of multi-granularity testing criteria named DeepGauge is presented in [219] that works by rendering a multi-faceted testbed. The security analysis of neural networks based system using symbolic intervals is presented in [220] which uses interval arithmetics and symbolic intervals together with other optimization methods to minimize confidence bound of overestimation of outputs. A coverage guided fuzzing method for testing neural networks for goals (e.g., finding numerical

[6]https://github.com/evaluating-adversarial-robustness/adv-eval-paper

[7]MC/DC (Modified Condition/Decision Coverage) is a method of measuring the extent to which safety-critical software has been adequately tested.

errors, generating disagreements, and determining the undesirable behavior of models) is presented in [221]. In [222], the first approach utilizing differential fuzzing testing is presented for exploiting incorrect behavior of DL systems.

2) Automated Testing of ML Empowered Autonomous Vehicles (An Overview): To ensure a completely secure functionality of autonomous vehicles in a real-world environment, the development of automated testing tools is required. As the backbone of autonomous vehicles leverage different ML techniques for building decision systems at different levels, e.g., perception, decision making, and control, etc. In this section, we provide an overview of various studies performing test of autonomous vehicles.

Tian *et al.* [223] proposed and investigated a tool named DeepTest to perform testing of DNN empowered autonomous vehicles to automatically detect erroneous behaviors of the vehicle that can potentially cause fatal accidents. Their proposed tool automatically generates test cases using changes in real-world road conditions such as weather and lighting conditions and then systematically explores different components of DNN logic that maximize the number of activated neurons. Furthermore, they tested three DNNs that won top positions in Udacity self-driving car challenge and found various erroneous behaviors in different real-world road conditions (e.g., rain, fog, blurring, etc.) that led to fatal accidents. In [224], used a GAN-based approach to generate synthetic scenes of different driving conditions for testing autonomous cars. A metamorphic testing approach for evaluating the software part of self-driving vehicles is presented in [225].

A generic framework for testing security and robustness of ML models for computer vision systems depicting realistic properties is presented in [130]. Authors evaluated the security of fifteen state of the art computer vision systems in black box setting including Nvidia's Dave self-driving system. Moreover, it has been provably demonstrated that there exists a trade-off between adversarial robustness to perturbations and the standard accuracy of the model in a fairly simple and natural setting [226]. A simulation-based framework for generating adversarial test cases to evaluate the closed-loop properties of ML enabled autonomous vehicles is presented in [170]. In [227], authors generated adversarial driving scenes using Bayesian optimization to improve self-driving behavior utilizing vision-based imitation learning. An autoencoder-based approach for automatic identification of unusual events using small dashcam video and the inertial sensor is presented in [228] that can potentially be used to develop a robust autonomous driving system. Various factors and challenges impacting driveability of autonomous vehicles along with an overview of available datasets for training self-driving is presented in [229] and challenges in designing such datasets are described in [230]. Furthermore, Dreossi *et al.* suggested that while robustifying the ML systems, the effect of adversarial ML should be studied by considering the semantics and context of the whole system [231]. For example, in DL empowered autonomous vehicle, not every adversarial observation might lead to harmful action(s). Moreover, one might be interested in those adversarial examples that can significantly modify the desired semantics of the whole system.

VI. Open Research Issues

The advancement of ML research and its state of the art performance in various complex domains, in particular, the advent of more sophisticated DL methods might be an inherent panacea to the conventional challenges of vehicular networks. However, ML/DL methods cannot be naively applied to vehicular networks that possess unique characteristics and adaption of these methods for learning such distinguishing features of vehicular networks is a challenging task [232]. In this section, we highlight a few promising areas of research that require further investigation.

A. Efficient Distributed Data Storage

In the connected vehicular ecosystem, the data is generated and stored in a distributed fashion that raises a question about the applicability of ML/DL models at a global level. As ML models are developed with the assumption that data is easily accessible and managed by a central entity, there is a need to utilize distributed learning methods for connected vehicles so that data may be scalably acquired from multiple units in the ecosystem.

B. Interpretable ML

Another major security vulnerability in CAVs is the lack of interpretability of ML schemes. ML techniques in general and DL techniques specifically are based on the idea of function approximation, where the approximation of the empirical function is performed using DNN architectures. Current works in ML/DL lack interpretability, which is resulting in a major hurdle in the progress of ML/DL empowered CAVs. The lack of interpretability is exploited by the adversaries to construct adversarial examples for fooling the deployed ML/DL schemes in autonomous vehicles, i.e., physical attacks on self-driving vehicles as discussed above. Development of secure, explainable, and interpretable ML techniques for security-critical applications of CAVs is another open research issue.

C. Defensive and Secure ML

Despite many defense proposals presented in the literature for adversarial attacks, developing adversarially robust ML models remains yet another open research problem. Almost every defense has been shown to be only effective for a specific attack type and fails for stronger or unseen attacks. Moreover, most defenses address the problem of adversarial attacks for computer vision tasks but adversarial ML is being developed for many other vertical application domains. Therefore, development of efficient and effective novel defense strategies is essentially required, particularly, for safety-critical applications, e.g., communication between connected vehicles.

D. Privacy Preserving ML

Preserving privacy in any user-centric application is of high concern. Privacy means that models should not reveal any additional information about the subjects involved in collected training data (a.k.a. differential privacy) [233]. As CAVs involve human subjects, ML model learning should be capable

of preserving the privacy of drivers, passengers, and pedestrians where privacy breaches can results in extremely harmful consequences.

E. Security Centric Proxy Metrics

Development of security-centric proxy metrics to evaluate security threats against systems is fundamentally important. Currently, there is no way to formalize different types of perturbation properties, e.g., indistinguishable and content-preserving, etc. In addition, there is no function to determine that a specific transformation is content-preserving. Similarly, the process of measuring perceptual similarity between two images is very complex and widely used perceptual metrics are shallow functions that fail to account for many subtle distinctions of human perception [234].

F. Fair and Accountable ML

The literature on ML reveals that ML-based results and predictions lack fairness and accountability. The *fairness* property ensures that the ML model did not nurture discrimination against specific cases, e.g., favoring cyclists over pedestrians. This bias in ML predictions is introduced by the biased training data and results in social bias and higher error rate for a particular demographic group. For example, researchers identified a risk of bias in the perception system of autonomous vehicles to recognize pedestrians with dark skin [235]. This is an experimental work in which authors evaluated different models developed by other academic researchers for autonomous vehicles. Despite the fact that this work does not use an actual object detection model that is being used by autonomous vehicles in the market, nor did it use the training data being used by autonomous vehicle manufactures, this study highlights a major vulnerability of ML models used in autonomous vehicles and raises serious concerns about their applicability in real-world settings where a self-driving vehicle may encounter people from a variety of demographic backgrounds.

The *accountability* of ML models is associated with their interpretability property as we are interested in developing such models that can explain their predictions using the models' internal parameters. The notion of accountability is fundamentally important to understand ML model failures for adversarial examples.

G. Robustifying ML Models Against Distribution Drifts

To restrict the integrity attacks, ML models should be made robust against distribution drifts which refer to the situation where train and test data distributions are different. This difference between the training and test distributions gives rise to adversarial examples. These examples can also be considered as the worst case distribution drifts [117]. It is fairly clear that the data collection process in the vehicular ecosystem is temporal and dynamic in nature so such distribution drifts are highly possible and will affect the robustness of the underlying ML systems. Moreover, such drifts can be exploited by the adversaries to create adversarial samples during inference, for example, in [236] authors investigated this distribution drift by introducing positively connotated words in spam emails to evade detection. Moreover, modification of the training distribution is also possible in a similar way and distribution drift violates the widely known presumption that we can achieve low learning error when a large training data is available. Ford *et al.* [237] have presented empirical and theoretical evidence that adversarial examples are a consequence of test error in noise caused by a distributional shift in the data. To ensure that the adversarial defense is trustworthy, it must provide defense against data distribution shifts. As the perception system of CAVs is mainly based on data-driven modeling using historical training data, it is highly susceptible to the problem of distribution drifts. Therefore, robustifying ML models against the aforementioned distribution drifts is very important. One way to counter this problem is to leverage deep reinforcement learning (RL) algorithms for developing the perception system of autonomous vehicles but this is not yet practically possible, as the state and action spaces in realistic settings (road and vehicular environment) are continuous and very complex. Therefore, fine control is required for the efficacy of the system [156]. However, the work on leveraging deep RL-based methods for autonomous vehicles is building up. For instance, Sallab *et al.* proposed a deep RL-based framework for autonomous vehicles that enables the vehicle to handle partially observable scenarios [238]. They investigated the effectiveness of their system using an open source 3D car racing simulator (TORCS)[8] and demonstrated that their model was able to learn complex road curvatures and simple inter-vehicle interactions. On the counter side, deep RL-based systems have been shown vulnerable to policy induction attacks [239].

VII. Conclusion

The recent discoveries that machine learning (ML) techniques are vulnerable to adversarial perturbations have raised questions on the security of connected and autonomous vehicles (CAVs), which utilize ML techniques for various tasks ranging from environmental perception to objection recognition and movement prediction. The safety-critical nature of CAVs clearly demands that the technology it uses should be robust to all kinds of potential security threats—be they accidental, intentional, or adversarial. In this work, we present for the first time a comprehensive analysis of the challenges posed by adversarial ML attacks for CAVs aggregating insights from both the ML and CAV literature. Our major contributions include: a broad description of the ML pipeline used in CAVs; description of the various adversarial attacks that can be launched on the various components of the CAV ML pipeline; a detailed taxonomy of the adversarial ML threat for CAVs; a comprehensive survey of adversarial ML attacks and defenses proposed in literature. Finally, open research challenges and future directions are discussed to provide readers with the opportunity to develop robust and efficient solutions for the application of ML models in CAVs.

[8]http://torcs.sourceforge.net/

REFERENCES

[1] S. E. Shladover, "Connected and automated vehicle systems: Introduction and overview," *J. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 190–200, 2018.

[2] R. Hussain and S. Zeadally, "Autonomous cars: Research results, issues and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1275–1313, 2nd Quart., 2019.

[3] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, "LTE for vehicular networking: A survey," *IEEE Commun. Mag.*, vol. 51, no. 5, pp. 148–157, May 2013.

[4] H. Peng, L. Liang, X. Shen, and G. Y. Li, "Vehicular communications: A network layer perspective," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1064–1078, Feb. 2019.

[5] S. Checkoway *et al.*, "Comprehensive experimental analyses of automotive attack surfaces," in *Proc. USENIX Security Symp.*, San Francisco, CA, USA, 2011, pp. 77–92.

[6] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Security Privacy (EuroS&P)*, 2016, pp. 372–387.

[7] K. Eykholt *et al.*, "Robust physical-world attacks on deep learning visual classification," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 1625–1634.

[8] C. Sitawarin, A. N. Bhagoji, A. Mosenia, M. Chiang, and P. Mittal, "DARTS: Deceiving autonomous cars with toxic signs," 2018. [Online]. Available: arXiv:1802.06430.

[9] C. Szegedy *et al.*, "Intriguing properties of neural networks," 2013. [Online]. Available: arXiv:1312.6199.

[10] M. N. Mejri, J. Ben-Othman, and M. Hamdi, "Survey on VANET security challenges and possible cryptographic solutions," *Veh. Commun.*, vol. 1, no. 2, pp. 53–66, 2014.

[11] J. Gardiner and S. Nagaraja, "On the security of machine learning in Malware C&C detection: A survey," *ACM Comput. Surveys*, vol. 49, no. 3, p. 59, 2016.

[12] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial attacks and defences: A survey," 2018. [Online]. Available: arXiv:1810.00069.

[13] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018.

[14] J. E. Siegel, D. C. Erb, and S. E. Sarma, "A survey of the connected vehicle landscape—Architectures, enabling technologies, applications, and development areas," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2391–2406, Aug. 2018.

[15] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 96, pp. 2805–2824, Sep. 2019.

[16] W. Wang, B. Tang, R. Wang, L. Wang, and A. Ye, "A survey on adversarial attacks and defenses in text," 2019. [Online]. Available: arXiv:1902.07285.

[17] R. M. Gandia *et al.*, "Autonomous vehicles: Scientometric and bibliometric review," *Transport Rev.*, vol. 39, no. 1, pp. 9–28, 2019.

[18] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, Soc. Autom. Eng., Warrendale, PA, USA, 2016.

[19] S. E. Shladover, "Roadway automation technology—Research needs," *Transp. Res. Rec.*, no. 1283, pp. 158–167, 1990.

[20] E. D. Dickmanns, "Vision for ground vehicles: History and prospects," *Int. J. Veh. Auton. Syst.*, vol. 1, no. 1, pp. 1–44, 2002.

[21] H.-B. Glathe, "Prometheus—A cooperative effort of the European automotive manufacturers," SAE, Warrendale, PA, USA, Rep. 942430, 1994.

[22] S. Tsugawa, T. Saito, and A. Hosaka, "Super smart vehicle system: AVCS related systems for the future," in *Proc. IEEE Intell. Veh. Symp.*, 1992, pp. 132–137.

[23] J. H. Rillings, "Automated highways: Cars that drive themselves in tight formation might alleviate the congestion now plaguing urban freeways," *Sci. Amer.*, vol. 277, no. 4, pp. 80–85, 1997.

[24] E. Uhlemann, "Introducing connected vehicles [connected vehicles]," *IEEE Veh. Technol. Mag.*, vol. 10, no. 1, pp. 23–31, Feb. 2015.

[25] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 289–299, May 2014.

[26] S. Tsugawa, S. Jeschke, and S. E. Shladover, "A review of truck platooning projects for energy savings," *IEEE Trans. Intell. Veh.*, vol. 1, no. 1, pp. 68–77, Mar. 2016.

[27] J. Lioris, R. Pedarsani, F. Y. Tascikaraoglu, and P. Varaiya, "Platoons of connected vehicles can double throughput in urban roads," *Transp. Res. C Emerg. Technol.*, vol. 77, pp. 292–305, Apr. 2017.

[28] R. Staszewski and H. Estl, "Making cars safer through technology innovation," Texas Instrum., Dallas, TX, USA, White Paper, 2013.

[29] K. Koscher *et al.*, "Experimental security analysis of a modern automobile," in *Proc. IEEE Symp. Security Privacy (SP)*, 2010, pp. 447–462.

[30] P. Kleberger, T. Olovsson, and E. Jonsson, "Security aspects of the in-vehicle network in the connected car," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2011, pp. 528–533.

[31] S. E. Shladover, C. Nowakowski, X.-Y. Lu, and R. Ferlis, "Cooperative adaptive cruise control: Definitions and operating concepts," *Transp. Res. Rec.*, vol. 2489, no. 1, pp. 145–152, 2015.

[32] I. A. Sumra, H. B. Hasbullah, and J.-L. B. AbManan, "Attacks on security goals (confidentiality, integrity, availability) in VANET: A survey," in *Vehicular Ad-Hoc Networks for Smart Cities*. Singapore: Springer, 2015, pp. 51–61.

[33] K. Lim, K. M. Tuladhar, and H. Kim, "Detecting location spoofing using ADAS sensors in VANETs," in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, 2019, pp. 1–4.

[34] S. Bittl, A. A. Gonzalez, M. Myrtus, H. Beckmann, S. Sailer, and B. Eissfeller, "Emerging attacks on VANET security based on GPS time spoofing," in *Proc. IEEE Conf. Commun. Netw. Security (CNS)*, 2015, pp. 344–352.

[35] J. Petit and S. E. Shladover, "Potential cyberattacks on automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 546–556, Apr. 2015.

[36] B. Parno and A. Perrig, "Challenges in securing vehicular networks," in *Proc. Workshop Hot Topics Netw. (HotNets-IV)*, 2005, pp. 1–6.

[37] B. Mokhtar and M. Azab, "Survey on security issues in vehicular ad hoc networks," *Alexandria Eng. J.*, vol. 54, no. 4, pp. 1115–1126, 2015.

[38] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (VANETS): Status, results, and challenges," *Telecommun. Syst.*, vol. 50, no. 4, pp. 217–241, 2012.

[39] J. Li, H. Lu, and M. Guizani, "ACPN: A novel authentication framework with conditional privacy-preservation and non-repudiation for VANETs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 938–948, Apr. 2015.

[40] J. R. Douceur, "The sybil attack," in *Proc. Int. Workshop Peer-to-Peer Syst.*, 2005, pp. 251–260.

[41] D. He, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 12, pp. 2681–2691, Dec. 2015.

[42] M. Amoozadeh *et al.*, "Security vulnerabilities of connected vehicle streams and their impact on cooperative driving," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 126–132, Jun. 2015.

[43] N. Lyamin, D. Kleyko, Q. Delooz, and A. Vinel, "Ai-based malicious network traffic detection in VANETs," *IEEE Netw.*, vol. 32, no. 6, pp. 15–21, Nov./Dec. 2018.

[44] S. Ucar, S. C. Ergen, and O. Ozkasap, "Data-driven abnormal behavior detection for autonomous platoon," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, 2017, pp. 69–72.

[45] G. K. Rajbahadur, A. J. Malton, A. Walenstein, and A. E. Hassan, "A survey of anomaly detection for connected vehicle cybersecurity and safety," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2018, pp. 421–426.

[46] M. T. Garip, M. E. Gursoy, P. Reiher, and M. Gerla, "Congestion attacks to autonomous cars using vehicular botnets," in *Proc. NDSS Workshop Security Emerg. Netw. Technol. (SENT)*, 2015, pp. 1–9.

[47] P. Koopman and M. Wagner, "Autonomous vehicle safety: An interdisciplinary challenge," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 1, pp. 90–96, Jan. 2017.

[48] Y. Shoukry, P. Martin, P. Tabuada, and M. Srivastava, "Non-invasive spoofing attacks for anti-lock braking systems," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, 2013, pp. 55–72.

[49] B. Wiedersheim, Z. Ma, F. Kargl, and P. Papadimitratos, "Privacy in inter-vehicular networks: Why simple pseudonym change is not enough," in *Proc. IEEE 7th Int. Conf. Wireless On Demand Netw. Syst. Services (WONS)*, 2010, pp. 176–183.

[50] S. Wang and Y. He, "A trust system for detecting selective forwarding attacks in VANETs," in *Proc. Int. Conf. Big Data Comput. Commun.*, 2016, pp. 377–386.

[51] K. A. Alheeti, A. Gruebler, and K. McDonald-Maier, "Intelligent intrusion detection of grey hole and rushing attacks in self-driving vehicular networks," *Computers*, vol. 5, no. 3, p. 16, 2016.

[52] M. Raya, D. Jungels, P. Papadimitratos, I. Aad, and J.-P. Hubaux, *Certificate Revocation in Vehicular Networks*, Lab. Comput. Commun. Appl., School Comput. Sci., EPFL, Lausanne, Switzerland, 2006.

[53] F. Cunha *et al.*, "Data communication in VANETs: Protocols, applications and challenges," *Ad Hoc Netw.*, vol. 44, pp. 90–103, Jul. 2016.

[54] J. J. Blum, A. Eskandarian, and L. J. Hoffman, "Challenges of intervehicle ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 347–351, Dec. 2004.

[55] R. D. R. Fontes, C. Campolo, C. E. Rothenberg, and A. Molinaro, "From theory to experimental evaluation: Resource management in software-defined vehicular networks," *IEEE Access*, vol. 5, pp. 3069–3076, 2017.

[56] L. Liang, H. Ye, and G. Y. Li, "Toward intelligent vehicular networks: A machine learning framework," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 124–135, Feb. 2019.

[57] H. Ye and G. Y. Li, "Deep reinforcement learning for resource allocation in V2V communications," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2018, pp. 1–6.

[58] X. Yang, L. Liu, N. H. Vaidya, and F. Zhao, "A vehicle-to-vehicle communication protocol for cooperative collision warning," in *Proc. 1st Annu. Int. Conf. Mobile Ubiquitous Syst. Netw. Services (MOBIQUITOUS)*, 2004, pp. 114–123.

[59] J. Wan, J. Liu, Z. Shao, A. Vasilakos, M. Imran, and K. Zhou, "Mobile crowd sensing for traffic prediction in Internet of Vehicles," *Sensors*, vol. 16, no. 1, p. 88, 2016.

[60] A. M. de Souza, R. S. Yokoyama, G. Maia, A. Loureiro, and L. Villas, "Real-time path planning to prevent traffic jam through an intelligent transportation system," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2016, pp. 726–731.

[61] M. Wang, H. Shan, R. Lu, R. Zhang, X. Shen, and F. Bai, "Real-time path planning based on hybrid-VANET-enhanced transportation system," *IEEE Trans. Veh. Technol.*, vol. 64, no. 5, pp. 1664–1678, May 2015.

[62] L. Yao, J. Wang, X. Wang, A. Chen, and Y. Wang, "V2X routing in a VANET based on the hidden Markov model," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 889–899, Mar. 2018.

[63] G. Xue, Y. Luo, J. Yu, and M. Li, "A novel vehicular location prediction based on mobility patterns for routing in urban VANET," *EURASIP J. Wireless Commun. Netw.*, vol. 2012, no. 1, p. 222, 2012.

[64] F. Zeng, R. Zhang, X. Cheng, and L. Yang, "Channel prediction based scheduling for data dissemination in VANETs," *IEEE Commun. Lett.*, vol. 21, no. 6, pp. 1409–1412, Jun. 2017.

[65] A. Karami, "ACCPNDN: Adaptive congestion control protocol in named data networking by learning capacities using optimized time-lagged feedforward neural network," *J. Netw. Comput. Appl.*, vol. 56, pp. 1–18, Oct. 2015.

[66] N. Taherkhani and S. Pierre, "Centralized and localized data congestion control strategy for vehicular ad hoc networks using a machine learning clustering algorithm," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 11, pp. 3275–3285, Nov. 2016.

[67] Z. Li, C. Wang, and C.-J. Jiang, "User association for load balancing in vehicular networks: An online reinforcement learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 8, pp. 2217–2228, Aug. 2017.

[68] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA)*, 2016, pp. 130–139.

[69] Q. Zheng, K. Zheng, H. Zhang, and V. C. M. Leung, "Delay-optimal virtualized radio resource scheduling in software-defined vehicular networks via stochastic learning," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 7857–7867, Oct. 2016.

[70] R. F. Atallah, C. M. Assi, and J. Y. Yu, "A reinforcement learning technique for optimizing downlink scheduling in an energy-limited vehicular network," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 4592–4601, Jun. 2017.

[71] R. Atallah, C. Assi, and M. Khabbaz, "Deep reinforcement learning-based scheduling for roadside communication networks," in *Proc. IEEE 15th Int. Symp. Model. Optim. Mobile Ad Hoc Wireless Netw. (WiOpt)*, 2017, pp. 1–8.

[72] B. D. Kim, C. M. Kang, J. Kim, S. H. Lee, C. C. Chung, and J. W. Choi, "Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, 2017, pp. 399–404.

[73] C. Yan, H. Xie, D. Yang, J. Yin, Y. Zhang, and Q. Dai, "Supervised hash coding with deep neural network for environment perception of intelligent vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 284–295, Jan. 2018.

[74] F. Rosique, P. J. Navarro, C. Fernández, and A. Padilla, "A systematic review of perception system and simulators for autonomous vehicles research," *Sensors*, vol. 19, no. 3, p. 648, 2019.

[75] C. Chen, A. Seff, A. L. Kornhauser, and J. Xiao, "DeepDriving: Learning affordance for direct perception in autonomous driving," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2722–2730.

[76] S. Ramos, S. K. Gehrig, P. Pinggera, U. Franke, and C. Rother, "Detecting unexpected obstacles for self-driving cars: Fusing deep learning and geometric modeling," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2017, pp. 1025–1032.

[77] R. O. Chavez-Garcia and O. Aycard, "Multiple sensor fusion and classification for moving object detection and tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 525–534, Feb. 2016.

[78] H. Wang, B. Wang, B. Liu, X. Meng, and G. Yang, "Pedestrian recognition and tracking using 3D LIDAR for autonomous vehicle," *Robot. Auton. Syst.*, vol. 88, pp. 71–78, Feb. 2017.

[79] Y. Zeng, X. Xu, D. Shen, Y. Fang, and Z. Xiao, "Traffic sign recognition using kernel extreme learning machines with deep perceptual features," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1647–1653, Jun. 2017.

[80] V. John, K. Yoneda, B. Qi, Z. Liu, and S. Mita, "Traffic light recognition in varying illumination using deep learning and saliency map," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, 2014, pp. 2286–2291.

[81] L. Lin, S. Gong, T. Li, and S. Peeta, "Deep learning-based human-driven vehicle trajectory prediction and its application for platoon control of connected and autonomous vehicles," in *Proc. Auton. Veh. Symp.*, 2018, p. 30.

[82] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2595–2621, 4th Quart., 2018.

[83] L. Ye and T. Yamamoto, "Modeling connected and autonomous vehicles in heterogeneous traffic flow," *Physica A Stat. Mech. Appl.*, vol. 490, pp. 269–277, Jan. 2018.

[84] F. Zhang, C. Martinez, C. Daniel, D. Cao, and A. C. Knoll, "Neural network based uncertainty prediction for autonomous vehicle application," *Front. Neurorobot.*, vol. 13, p. 12, May 2019.

[85] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transp. Res. C Emerg. Technol.*, vol. 60, pp. 416–442, Nov. 2015.

[86] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, "Vehicle trajectory prediction based on motion model and maneuver recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 4363–4369.

[87] T. Li, "Modeling uncertainty in vehicle trajectory prediction in a mixed connected and autonomous vehicle environment using deep learning and kernel density estimation," in *Proc. 4th Annu. Symp. Transp. Informat.*, 2018.

[88] M. Bojarski *et al.*, "End to end learning for self-driving cars," 2016. [Online]. Available: arXiv:1604.07316.

[89] Z. Chen and X. Huang, "End-to-end learning for lane keeping of self-driving cars," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2017, pp. 1856–1860.

[90] J. Liu, P. Hou, L. Mu, Y. Yu, and C. Huang, "Elements of effective deep reinforcement learning towards tactical driving decision making," 2018. [Online]. Available: arXiv:1802.00332.

[91] M. Bouton, J. Karlsson, A. Nakhaei, K. Fujimura, M. J. Kochenderfer, and J. Tumova, "Reinforcement learning with probabilistic guarantees for autonomous driving," 2019. [Online]. Available: arXiv:1904.07189.

[92] Y. Zhang, P. Sun, Y. Yin, L. Lin, and X. Wang, "Human-like autonomous vehicle speed control by deep reinforcement learning with double Q-learning," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2018, pp. 1251–1256.

[93] Y. He, N. Zhao, and H. Yin, "Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 44–55, Jan. 2018.

[94] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, "Cooperative adaptive cruise control in real traffic situations," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 296–305, Feb. 2014.

[95] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PLoS ONE*, vol. 11, no. 6, 2016, Art. no. e0155781.

[96] D. G. Yang *et al.*, "Intelligent and connected vehicles: Current status and future perspectives," *Sci. China Technol. Sci.*, vol. 61, no. 10, pp. 1446–1471, 2018.

[97] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 1807–1814.

[98] B. Biggio *et al.*, "Evasion attacks against machine learning at test time," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Disc. Databases*, 2013, pp. 387–402.

[99] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014. [Online]. Available: arXiv:1412.6572.

[100] A. M. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 427–436.

[101] E. B. Khalil, A. Gupta, and B. Dilkina, "Combinatorial attacks on binarized neural networks," 2018. [Online]. Available: arXiv:1810.03538.

[102] E. R. Balda, A. Behboodi, and R. Mathar, "On generation of adversarial examples using convex programming," in *Proc. 52nd Asilomar Conf. Signals Syst. Comput.*, 2018, pp. 60–65.

[103] E. Wong and Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 5283–5292.

[104] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, Oct. 2019.

[105] J. Gilmer, R. P. Adams, I. J. Goodfellow, D. Andersen, and G. E. Dahl, "Motivating the rules of the game for adversarial example research," 2018. [Online]. Available: arXiv:1807.06732.

[106] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[107] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Rep. TR-2009, 2009.

[108] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[109] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[110] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.

[111] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[112] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.

[113] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[114] K. Riggle. (2019). *An Introduction to Approachable Threat Modeling*. Accessed: Apr. 23, 2019. [Online]. Available: https://increment.com/security/approachable-threat-modeling/

[115] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, "The security of machine learning," *Mach. Learn.*, vol. 81, no. 2, pp. 121–148, 2010.

[116] F. Tramèr, N. Papernot, I. J. Goodfellow, D. Boneh, and P. McDaniel, "The space of transferable adversarial examples," 2017. [Online]. Available: arXiv:1704.03453.

[117] N. Papernot, P. McDaniel, A. Sinha, and M. Wellman, "Towards the science of security and privacy in machine learning," 2016. [Online]. Available: arXiv:1611.03814.

[118] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial Intelligence Safety and Security*. San Rafael, CA, USA: CRC Press, 2018, pp. 99–112.

[119] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2574–2582.

[120] N. Carlini and D. A. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Security Privacy (SP)*, 2017, pp. 39–57.

[121] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1765–1773.

[122] S. Sarkar, A. Bansal, U. Mahbub, and R. Chellappa, "Upset and ANGRI: Breaking high performance image classifiers," 2017. [Online]. Available: arXiv:1707.01159.

[123] M. M. Cisse, Y. Adi, N. Neverova, and J. Keshet, "Houdini: Fooling deep structured visual and speech recognition models with adversarial examples," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6977–6987.

[124] S. Baluja and I. Fischer, "Learning to attack: Adversarial transformation networks," in *Proc. AAAI*, 2018, pp. 2687–2695.

[125] N. Dalvi *et al.*, "Adversarial classification," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2004, pp. 99–108.

[126] D. Lowd and C. Meek, "Adversarial learning," in *Proc. 11th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2005, pp. 641–647.

[127] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Proc. ACM Symp. Inf. Comput. Commun. Security*, 2006, pp. 16–25.

[128] L. Huang, A. D. Joseph, B. Nelson, B. I. P. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proc. 4th ACM Workshop Security Artif. Intell.*, 2011, pp. 43–58.

[129] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry, "A rotation and a translation suffice: Fooling CNNs with simple transformations," 2017. [Online]. Available: arXiv:1712.02779.

[130] K. Pei, Y. Cao, J. Yang, and S. Jana, "Towards practical verification of machine learning: The case of computer vision systems," 2017. [Online]. Available: arXiv:1712.01785.

[131] Y. Liu *et al.*, "Trojaning attack on neural networks," in *Proc. 25th Annu. Netw. Distrib. Syst. Security Symp. (NDSS)*, Feb. 2018, pp. 1–15.

[132] N. Papernot, P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Security*, 2017, pp. 506–519.

[133] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proc. 10th ACM Workshop Artif. Intell. Security*, 2017, pp. 3–14.

[134] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *Proc. IEEE Security Privacy Workshops (SPW)*, 2018, pp. 1–7.

[135] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS)*, 2017.

[136] Y. Vorobeychik and M. Kantarcioglu, "Adversarial machine learning," *Synth. Lectures Artif. Intell. Mach. Learn.*, vol. 12, no. 3, pp. 1–169, 2018.

[137] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognit.*, vol. 84, pp. 317–331, Dec. 2018.

[138] C. Sitawarin, A. N. Bhagoji, A. Mosenia, P. Mittal, and M. Chiang, "Rogue signs: Deceiving traffic sign recognition with malicious ADS and logos," 2018. [Online]. Available: arXiv:1801.02780.

[139] A. Arnab, O. Miksik, and P. H. S. Torr, "On the robustness of semantic segmentation models to adversarial attacks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 888–897.

[140] J. H. Metzen, M. C. Kumar, T. Brox, and V. Fischer, "Universal adversarial perturbations against semantic image segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2774–2783.

[141] V. Fischer, M. C. Kumar, J. H. Metzen, and T. Brox, "Adversarial examples for semantic image segmentation," 2017. [Online]. Available: arXiv:1703.01101.

[142] E. Tretschk, S. J. Oh, and M. Fritz, "Sequential attacks on agents for long-term adversarial goals," 2018. [Online]. Available: arXiv:1805.12487.

[143] A. Pattanaik, Z. Tang, S. Liu, G. Bommannan, and G. Chowdhary, "Robust deep reinforcement learning with adversarial attacks," in *Proc. 17th Int. Conf. Auton. Agents MultiAgent Syst.*, 2018, pp. 2040–2042.

[144] Y.-C. Lin *et al.*, "Tactics of adversarial attack on deep reinforcement learning agents," 2017. [Online]. Available: arXiv:1703.06748.

[145] J. Kos, I. Fischer, and D. Song, "Adversarial examples for generative models," in *Proc. IEEE Security Privacy Workshops (SPW)*, 2018, pp. 36–42.

[146] G. Gondim-Ribeiro, P. Tabacof, and E. Valle, "Adversarial attacks on variational autoencoders," 2018. [Online]. Available: arXiv:1806.04646.

[147] D. Pasquini, M. Mingione, and M. Bernaschi, "Out-domain examples for generative models," 2019. [Online]. Available: arXiv:1903.02926.

[148] M. Sato, J. Suzuki, H. Shindo, and Y. Matsumoto, "Interpretable adversarial perturbation in input embedding space for text," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2018, pp. 4323–4330.

[149] T. Miyato, A. M. Dai, and I. J. Goodfellow, "Adversarial training methods for semi-supervised text classification," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–11.

[150] J. Li, S. Ji, T. Du, B. Li, and T. Wang, "TextBugger: Generating adversarial text against real-world applications," in *Proc. Netw. Distrib. Syst. Security Symp. (NDSS)*, 2019.

[151] N. Papernot, P. McDaniel, A. Swami, and R. Harang, "Crafting adversarial input sequences for recurrent neural networks," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, 2016, pp. 49–54.

[152] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," in *Proc. Conf. Empirical Methods Nat. Lang. Process. (EMNLP)*, 2017, pp. 2021–2031.

[153] P. K. Mudrakarta, A. Taly, M. Sundararajan, and K. Dhamdhere, "Did the model understand the question?" in *Proc. 56th Annu. Meeting Assoc. Comput. Linguist.*, 2018, pp. 1896–1906.

[154] I. Corona, G. Giacinto, and F. Roli, "Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues," *Inf. Sci.*, vol. 239, pp. 201–225, Aug. 2013.

[155] Z. Lin, Y. Shi, and Z. Xue, "IDSGAN: Generative adversarial networks for attack generation against intrusion detection," 2018. [Online]. Available: arXiv:1809.02077.

[156] Z. Wang, "Deep learning-based intrusion detection with adversaries," *IEEE Access*, vol. 6, pp. 38367–38384, 2018.

[157] M. Salem, S. Taheri, and J. S. Yuan, "Anomaly generation using generative adversarial networks in host based intrusion detection," 2018. [Online]. Available: arXiv:1812.04697.

[158] G. Wang, T. Wang, H. Zheng, and B. Y. Zhao, "Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers," in *Proc. 23rd USENIX Security Symp. (USENIX Security)*, 2014, pp. 239–254.

[159] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial perturbations against deep neural networks for malware classification," 2016. [Online]. Available: arXiv:1606.04435.

[160] B. Kolosnjaji *et al.*, "Adversarial malware binaries: Evading deep learning for malware detection in executables," in *Proc. IEEE 26th Eur. Signal Process. Conf. (EUSIPCO)*, 2018, pp. 533–537.

[161] M. Usama, J. Qadir, and A. Al-Fuqaha, "Adversarial attacks on cognitive self-organizing networks: The challenge and the way forward," in *Proc. IEEE 43rd Conf. Local Comput. Netw. Workshops (LCN Workshops)*, 2018, pp. 90–97.

[162] M. E. Ahmed and H. Kim, "Poster: Adversarial examples for classifiers in high-dimensional network data," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2017, pp. 2467–2469.

[163] E. Viegas, A. Santin, V. Abreu, and L. S. Oliveira, "Stream learning and anomaly-based intrusion detection in the adversarial settings," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2017, pp. 773–778.

[164] L. Schönherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, "Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding," 2018. [Online]. Available: arXiv:1808.05665.

[165] S.-T. Chen, C. Cornelius, J. Martin, and D. H. P. Chau, "ShapeShifter: Robust physical adversarial attack on faster R-CNN object detector," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Disc. Databases*, 2018, pp. 52–68.

[166] H. Zhou *et al.*, "DeepBillboard: Systematic physical-world testing of autonomous driving systems," 2018. [Online]. Available: arXiv:1812.10812.

[167] Y. Zhang, H. Foroosh, P. David, and B. Gong, "CAMOU: Learning physical vehicle camouflages to adversarially attack detectors in the wild," in *Proc. ICLR*, 2018, pp. 1–20.

[168] D. Song *et al.*, "Physical adversarial examples for object detectors," in *Proc. 12th USENIX Workshop Offensive Technol. (WOOT)*, 2018, p. 1.

[169] K. Eykholt *et al.*, "Robust physical-world attacks on deep learning visual classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1625–1634.

[170] C. E. Tuncali, G. Fainekos, H. Ito, and J. Kapinski, "Simulation-based adversarial test generation for autonomous vehicles with machine learning components," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2018, pp. 1555–1562.

[171] S. Yaghoubi and G. Fainekos, "Gray-box adversarial testing for control systems with machine learning components," in *Proc. 22nd ACM Int. Conf. Hybrid Syst. Comput. Control (HSCC)*, 2019, pp. 179–184.

[172] A. Boloor, X. He, C. Gill, Y. Vorobeychik, and X. Zhang, "Simple physical adversarial examples against end-to-end autonomous driving models," 2019. [Online]. Available: arXiv:1903.05157.

[173] A. M. Aung, Y. Fadila, R. Gondokaryono, and L. Gonzalez, "Building robust deep neural networks for road sign detection," 2017. [Online]. Available: arXiv:1712.09327.

[174] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.

[175] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018. [Online]. Available: arXiv:1804.02767.

[176] M. Bansal, A. Krizhevsky, and A. Ogale, "ChauffeurNet: Learning to drive by imitating the best and synthesizing the worst," 2018. [Online]. Available: arXiv:1812.03079.

[177] R. Huang, B. Xu, D. Schuurmans, and C. Szepesvári, "Learning with a strong adversary," 2015. [Online]. Available: arXiv:1511.03034.

[178] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient SMT solver for verifying deep neural networks," in *Proc. Int. Conf. Comput.-Aided Verification*, 2017, pp. 97–117.

[179] S. Gu and L. Rigazio, "Towards deep neural network architectures robust to adversarial examples," 2014. [Online]. Available: arXiv:1412.5068.

[180] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," in *Proc. 25th Annu. Netw. Distrib. Syst. Security Symp. (NDSS)*, Feb. 2018.

[181] W. He, J. Wei, X. Chen, N. Carlini, and D. Song, "Adversarial example defense: Ensembles of weak defenses are not strong," in *Proc. 11th USENIX Workshop Offensive Technol. (WOOT)*, 2017, pp. 1–11.

[182] J. Gao, B. Wang, Z. Lin, W. Xu, and Y. Qi, "DeepCloak: Masking deep neural network models for robustness against adversarial samples," 2017. [Online]. Available: arXiv:1702.06763.

[183] S. Garg, V. Sharan, B. Zhang, and G. Valiant, "A spectral view of adversarially robust features," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 10159–10169.

[184] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "PixelDefend: Leveraging generative models to understand and defend against adversarial examples," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–20.

[185] G. Jin, S. Shen, D. Zhang, F. Dai, and Y. Zhang, "APE-GAN: Adversarial perturbation elimination with GAN," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2019, pp. 3842–3846.

[186] D. Meng and H. Chen, "MAGNET: A two-pronged defense against adversarial examples," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2017, pp. 135–147.

[187] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE Symp. Security Privacy (SP)*, 2016, pp. 582–597.

[188] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. Deep Learn. Workshop Adv. Neural Inf. Process. Syst. (NIPS)*, 2014.

[189] N. Carlini, G. Katz, C. Barrett, and D. L. Dill, "Ground-truth adversarial examples," 2017. [Online]. Available: arXiv:1709.10207.

[190] A. S. Ross and F. Doshi-Velez, "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1660–1669.

[191] C. Lyu, K. Huang, and H.-N. Liang, "A unified gradient regularization family for adversarial examples," in *Proc. IEEE Int. Conf. Data Min.*, 2015, pp. 301–309.

[192] J. Bradshaw, A. G. de G. Matthews, and Z. Ghahramani, "Adversarial examples, uncertainty, and transfer testing robustness in Gaussian process hybrid deep networks," 2017. [Online]. Available: arXiv:1707.02476.

[193] L. Schott, J. Rauber, M. Bethge, and W. Brendel, "Towards the first adversarially robust neural network model on MNIST," in *Proc. 7th Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–17.

[194] G. Tao, S. Ma, Y. Liu, and X. Zhang, "Attacks meet interpretability: Attribute-steered detection of adversarial samples," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7728–7739.

[195] N. Carlini, "Is AmI (attacks meet interpretability) robust to adversarial examples?" 2019. [Online]. Available: arXiv:1902.02322.

[196] L. Nguyen, S. Wang, and A. Sinha, "A learning and masking approach to secure learning," in *Proc. Int. Conf. Decis. Game Theory Security*, 2018, pp. 453–464.

[197] J. Lu, T. Issaranon, and D. Forsyth, "SafetyNet: Detecting and rejecting adversarial examples robustly," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 446–454.

[198] D. Gopinath, G. Katz, C. S. Pasareanu, and C. Barrett, "DeepSafe: A data-driven approach for checking adversarial robustness in neural networks," 2017. [Online]. Available: arXiv:1710.00486.

[199] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–12.

[200] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Towards proving the adversarial robustness of deep neural networks," in *Proc. Formal Verification Auton. Veh. (FVAV) Workshop*, 2017, pp. 19–26.
[201] F. Tramer, A. Kurakin, N. Papernot, I. J. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–20.
[202] G. K. Santhanam and P. Grnarova, "Defending against adversarial attacks by leveraging an entire GAN," 2018. [Online]. Available: arXiv:1805.10652.
[203] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–17.
[204] A. Prakash, N. Moran, S. Garber, A. DiLillo, and J. Storer, "Deflecting adversarial attacks with pixel deflection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8571–8580.
[205] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–16.
[206] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–12.
[207] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1778–1787.
[208] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–23.
[209] J. Buckman, A. Roy, C. Raffel, and I. J. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–22.
[210] G. S. Dhillon *et al.*, "Stochastic activation pruning for robust adversarial defense," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–13.
[211] F. Croce, M. Andriushchenko, and M. Hein, "Provable robustness of ReLU networks via maximization of linear regions," in *Proc. Mach. Learn. Res.*, vol. 89, Apr. 2019, pp. 2057–2066.
[212] N. Carlini *et al.*, "On evaluating adversarial robustness," 2019. [Online]. Available: arXiv:1902.06705.
[213] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, p. 484, 2016.
[214] P. Rajpurkar *et al.*, "ChexNet: Radiologist-level pneumonia detection on chest x-rays with deep learning," 2017. [Online]. Available: arXiv:1711.05225.
[215] M. Grewal, M. M. Srivastava, P. Kumar, and S. Varadarajan, "RadNet: Radiologist level accuracy using deep learning for hemorrhage detection in CT scans," in *Proc. IEEE 15th Int. Symp. Biomed. Imag. (ISBI)*, 2018, pp. 281–284.
[216] N. Papernot, P. McDaniel, and I. J. Goodfellow, "Transferability in machine learning: From phenomena to black-box attacks using adversarial samples," 2016. [Online]. Available: arXiv:1605.07277.
[217] W. Ruan *et al.*, "Global robustness evaluation of deep neural networks with provable guarantees for L_0 norm," 2018. [Online]. Available: arXiv:1804.05805.
[218] Y. Sun, M. Wu, W. Ruan, X. Huang, M. Kwiatkowska, and D. Kroening, "Concolic testing for deep neural networks," in *Proc. 33rd ACM/IEEE Int. Conf. Autom. Softw. Eng. (ASE)*, 2018, pp. 109–119.
[219] L. Ma *et al.*, "DeepGauge: Multi-granularity testing criteria for deep learning systems," in *Proc. 33rd ACM/IEEE Int. Conf. Autom. Softw. Eng.*, 2018, pp. 120–131.
[220] S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana, "Formal security analysis of neural networks using symbolic intervals," in *Proc. 27th USENIX Security Symp. (USENIX Security)*, 2018, pp. 1599–1614.
[221] A. Odena, C. Olsson, D. Andersen, and I. J. Goodfellow, "TensorFuzz: Debugging neural networks with coverage-guided fuzzing," in *Proc. 36th Int. Conf. Mach. Learn.*, vol. 96, Jun. 2019, pp. 4901–4911.
[222] J. Guo, Y. Jiang, Y. Zhao, Q. Chen, and J. Sun, "DLFuzz: Differential fuzzing testing of deep learning systems," in *Proc. 26th ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, 2018, pp. 739–743.
[223] Y. Tian, K. Pei, S. Jana, and B. Ray, "DeepTest: Automated testing of deep-neural-network-driven autonomous cars," in *Proc. 40th Int. Conf. Softw. Eng. (ICSE)*, 2018, pp. 303–314.
[224] M. Zhang, Y. Zhang, L. Zhang, C. Liu, and S. Khurshid, "DeepRoad: GAN-based metamorphic testing and input validation framework for autonomous driving systems," in *Proc. 33rd ACM/IEEE Int. Conf. Autom. Softw. Eng.*, 2018, pp. 132–142.
[225] Z. Q. Zhou and L. Sun, "Metamorphic testing of driverless cars," *Commun. ACM*, vol. 62, no. 3, pp. 61–67, Feb. 2019.
[226] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness May be at odds with accuracy," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, vol. 1050, 2019, p. 11.
[227] Y. Abeysirigoonawardena, F. Shkurti, and G. Dudek, "Generating adversarial driving scenarios in high-fidelity simulators," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 8271–8277.
[228] H. Li, H. Wang, L. Liu, and M. Gruteser, "Automatic unusual driving event identification for dependable self-driving," in *Proc. 16th ACM Conf. Embedded Netw. Sensor Syst.*, 2018, pp. 15–27.
[229] J. Guo, U. Kurup, and M. Shah, "Is it safe to drive? An overview of factors, challenges, and datasets for driveability assessment in autonomous driving," 2018. [Online]. Available: arXiv:1811.11277.
[230] M. Uricar, D. Hurych, P. Krizek, and S. Yogamani, "Challenges in designing datasets and validation for autonomous driving," in *Proc. 14th Int. Conf. Comput. Vis. Theory Appl. (VISAPP)*, 2019, pp. 653–659.
[231] T. Dreossi, S. Jha, and S. A. Seshia, "Semantic adversarial deep learning," in *Proc. Int. Conf. Comput.-Aided Verification*, 2018, pp. 3–26.
[232] H. Ye, L. Liang, G. Y. Li, J. B. Kim, L. Lu, and M. Wu, "Machine learning for vehicular networks: Recent advances and application examples," *IEEE Veh. Technol. Mag.*, vol. 13, no. 2, pp. 94–101, Jun. 2018.
[233] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends® Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.
[234] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 586–595.
[235] B. Wilson, J. Hoffman, and J. Morgenstern, "Predictive inequity in object detection," 2019. [Online]. Available: arXiv:1902.11097.
[236] D. Lowd and C. Meek, "Good word attacks on statistical spam filters," in *Proc. 2nd Conf. Email Anti Spam (CEAS)*, vol. 2005, 2005.
[237] N. Ford, J. Gilmer, N. Carlini, and D. Cubuk, "Adversarial examples are a natural consequence of test error in noise," 2019. [Online]. Available: arXiv:1901.10513.
[238] A. E. L. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *Electron. Imag. Auton. Veh. Mach.*, vol. 2017, no. 19, pp. 70–76, 2017. [Online]. Available: https://doi.org/10.2352/ISSN.2470-1173.2017.19.AVM-023
[239] V. Behzadan and A. Munir, "Vulnerability of deep reinforcement learning to policy induction attacks," in *Proc. Int. Conf. Mach. Learn. Data Min. Pattern Recognit.*, 2017, pp. 262–275.

Adnan Qayyum received the bachelor's degree in electrical (computer) engineering from the COMSATS Institute of Information Technology, Wah, Pakistan, in 2014, and the M.S. degree in computer engineering (signal and image processing) from the University of Engineering and Technology, Taxila, Pakistan, in 2016. He is currently pursuing the Ph.D. degree in computer science with Information Technology University, Lahore, Pakistan. His research interests include autonomous vehicles, healthcare, and deep/machine learning.

Muhammad Usama received the bachelor's degree in telecommunication engineering from Government College University, Faisalabad, Pakistan, in 2010, and the master's degree from the National University of Computer and Emerging Sciences, Islamabad. He is currently pursuing the Ph.D. degree in electrical engineering from Information Technology University, Lahore, Pakistan. His research interests include adversarial machine learning and computer networks.

Junaid Qadir (Senior Member, IEEE) is the Director of the IHSAN—ICTD; Human Development; Systems; Big Data Analytics; Networks—Research Lab and the Chairperson of the Electrical Engineering Department, Information Technology University, Lahore, Pakistan. He has published more than 100 peer-reviewed articles at various high-quality research venues including more than 50 impact-factor journal publications at top international research journals, including *IEEE Communication Magazine*, the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATION, the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and the IEEE TRANSACTIONS ON MOBILE COMPUTING. His primary research interests are in the areas of computer systems and networking, applied machine learning, using ICT for development, and engineering education. He was awarded the Highest National Teaching Award in Pakistan the Higher Education Commissions Best University Teacher Award for the year from 2012 to 2013. He has been appointed as an ACM Distinguished Speaker for a three-year term starting from 2020. He is a Senior Member of ACM.

Ala Al-Fuqaha (Senior Member, IEEE) received the Ph.D. degree in computer engineering and networking from the University of Missouri–Kansas City, Kansas City, MO, USA. He is currently a Professor with the Information and Computing Technology Division, College of Science and Engineering, Hamad Bin Khalifa University, and the Computer Science Department, Western Michigan University. His research interests include the use of machine learning in general and deep learning in particular in support of the data-driven and self-driven management of large-scale deployments of IoT and smart city infrastructure and services, wireless vehicular networks, cooperation and spectrum access etiquette in cognitive radio networks, and management and planning of software defined networks. He serves on editorial boards of multiple journals, including the IEEE COMMUNICATIONS LETTER and *IEEE Network Magazine*. He also served as the Chair, the Co-Chair, and the Technical Program Committee Member of multiple international conferences including IEEE VTC, IEEE Globecom, IEEE ICC, and IWCMC. He is a Senior Member of ABET Program Evaluator.