IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 17, 2021, accepted March 24, 2021, date of publication March 31, 2021, date of current version April 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3070023

BCOOL: A Novel Blockchain Congestion Control Architecture Using Dynamic Service Function Chaining and Machine Learning for Next Generation Vehicular Networks

SAIDA MAAROUFI, (Senior Member, IEEE), AND SAMUEL PIERRE, (Senior Member, IEEE)
Department of Computer and Software Engineering, Ecole Polytechnique de Montreal, University of Montreal, Montreal, QC H3T 1J4, Canada
Corresponding author: Saida Maaroufi (s.maaroufi@polymtl.ca)

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

ABSTRACT This paper presents the first, novel, dynamic, resilient, and consistent Blockchain COngestion ContrOL (BCOOL) system for vehicular networks that fills the gap of trustworthy Blockchain congestion prediction systems. BCOOL relies on the heterogeneity of Machine Learning, Software-Defined Networks and Network Function Virtualization that is customized in three hybrid cloud/edge-based On/Offchain smart contract modules and ruled by an efficient and reliable communication protocol. BCOOL's first novel module aims at managing message and vehicle trustworthiness using a novel, dynamic and hybrid Blockchain Fog-based Distributed Trust Contract Strategy (FDTCS). The second novel module accurately and proactively predicts the occurrence of congestion, ahead of time, using a novel Hybrid On/Off-Chain Multiple Linear Regression Software-defined Contract Strategy (HOMLRCS). This module presents a virtualization facility layer to the third novel K-means/Random Forest-based On/Off-Chain Dynamic Service Function Chaining Contract Strategy (KRF-ODSFCS) that dynamically, securely and proactively predicts VNF placements and their chaining order in the context of SFCs w.r.t users' dynamic QoS priority demands. BCOOL exhibits a linear complexity and a strong resilience to failures. Simulation results show that BCOOL outperforms the next best comparable strategies by 80% and 100% reliability and efficiency gains in challenging data congestion environments. This yields to fast, reliable and accurate congestion prediction decisions, ahead of time, and optimizes transaction validation processing time. Globally, the Byzantine resilience, complexity and attack mitigation strategies along with simulation results prove that BCOOL securely predicts the congestion and provides real-time monitoring, fast and accurate SFC deployment decisions while lowering both capital and operational expenditures (CAPEX/OPEX) costs.

INDEX TERMS Blockchain, congestion prediction, random forest, K-means, machine learning (ML), network function virtualization (NFV), software-defined networks (SDN), quality of service (QoS), VANETs.

I. INTRODUCTION

Traffic congestion is unavoidable and is the root cause of road rage and huge delays. Approximately 1.35 million people die every year as a consequence of road traffic accidents directly linked to traffic congestion [69], [70].

Vehicular Adhoc Networks (VANets) was developed to provide safety and reduce congestion and road accidents by sending warning messages using vehicular applications based on congestion control protocols [17]. Vehicular applications are classified into two categories: 1) safety applications (e.g. collision and security distance warning) that aim at increasing drivers' awareness to emergency situations and nearby accidents, and 2) non-safety applications (e.g. real-time interactive games) that aim at providing comfortable and entertaining services to the drivers [16]–[18]. The huge need of those applications and congestion control protocols to deal with traffic congestion in vehicular environments opens up many challenges and requirements [17], [24]. In fact, the large amount of data, that represents dynamic user needs, and which is exchanged among vehicles and Road Side Units (RSUs) causes tremendous data congestion issues at different network levels [17]. This is due to several reasons and mostly to the unpredictable nature

The associate editor coordinating the review of this manuscript and approving it for publication was Guangjie Han.

of VANETs that drastically impacts the time-sensitive data congestion control delivery performance. Those traditional congestion control protocols waste a considerable amount of resources to continuously avoid data congestion without raising the importance of predicting the occurrence of congestion [20], [33], [45]. The performance of those protocols spontaneously and extremely degrades due to their entire negligence of a fundamental aspect of vehicular environments that is security [13], [14], [20]. Failure to tackle those issues along with the increasing, heterogenous and dynamic demands of traffic applications with high QoS requirements increase the vulnerability of both users and the network to diverse security attacks and ossify the whole networking infrastructure. This network ossification exacerbates the CAPEX and OPEX costs [8], [38]. Several research works [3], [34], [40] proposed NFV-based approaches to cope with network ossification. However, those solutions often require time to converge which does not fit with the urgent user requirements in terms of critical safety and emergency applications. Other works proposed Blockchain-based approaches for secure data dissemination [4]. However, they are still in their initial stages of research.

In this paper, we fill the gap of trustworthy Blockchain congestion prediction systems. We present the first, novel, dynamic, resilient and consistent Blockchain Congestion Control System (BCOOL) that relies on the heterogeneity of ML, SDN and NFV paradigms to reliably, immutably, and accurately predict the occurrence of congestion, at the edge of the network, and provide fast and accurate SFC deployment prediction decisions w.r.t vehicles' heterogenous QoS priority demands. BCOOL dynamically records vehicle, message trustworthiness and prediction transactions in a distributed ledger. It relies on three novel modules:

- The first novel hybrid Blockchain Fog-based Distributed Trust Contract Strategy (FDTCS) dynamically and securely manages message and vehicle trustworthiness, at the edge of the network to prevent distrustful users from acquiring access to the infrastructure;
- The second novel Hybrid software-defined On/Off-Chain-based Multiple Linear Regression Contract Strategy (HOMLRCS) relies on the output of FDTCS to accurately and reliably predict the occurrence of congestion, ahead of time, while optimizing the Blockchain validation consensus process;
- HOMLRCS presents a virtualization facility layer that introduces the third novel K-means/Random Forest-based Hybrid On/Off-Chain Dynamic Service Function Chaining Contract Strategy (KRF-ODSFCS). This strategy proactively predicts VNF placements and their chaining order and provides fast SFC deployment decisions that match priority user needs while lowering both CAPEX and OPEX costs.

The complexity analysis and attack mitigation strategies along with the Byzantine resilience and experimental results prove the robustness and effectiveness of BCOOL.

The rest of this paper is organized as follows. In Section II, we present some important background concepts and discuss the related literature. In Section III, we outline our problem. In Section IV, we lay out the BCOOL system along with its three main modules and describe its failure model. In Section V, we discuss BCOOL attack mitigation strategies while in Section VI we present and analyze BCOOL simulation results. Finally, Section VII concludes the paper.

II. BACKGROUND & RELATED WORK

Hereafter, we overview Blockchain protocols and architectures that have been devised for VANETs along with data congestion protocols and architectures for vehicular networks.

A. BLOCKCHAIN ARCHITECTURES FOR VANETs

Blockchain is a decentralized chain of blocks in which data is stored as a collection of transactions. For a block to be added to the chain, a transaction must happen and should be verified by a decentralized network of nodes. This network confirms the details of the transaction including the participants, the time and the amount. Then, the transaction is stored in a block with digital signatures and joins other transactions [83]. The block is then given a hash of the most recent block added to the Blockchain and can then be added to the Blockchain. Blockchain tackled the issue of data trust and implemented consensus models for users who want to join and add blocks to the chain [10], [82]. PoA is the consensus model that we deploy in this work.

In the PoA process, transactions and blocks are validated by verified nodes also called validators without solving complex mathematical problems. Blockchain nodes earn the qualification to become validators, so they have an incentive to retain the position they have acquired. Malicious validators are kicked out by the votes of validators. A PoA Blockchain is powerful than PoS and PoW because it is cheaper and safer [82], [83].

By investigating the literature, we find out that few research proposals tackle VANET challenges using the Blockchain technology. Most existing and recent research works that can be related to our work focus on Blockchain architectures/protocols for WSNs and IoT networks [5], [9], [12], [32]. On the other hand, research proposals, closest to our work, focused on the Blockchain computational overhead problem for resource constrained IoT devices [26]–[30]. However, those works are still in their initial stages of research and lack details regarding their specific features. Lu *et al.* [7] proposed a simple incentive mechanism based on a game theoretic approach that allows the execution of complex programs off the Blockchain to prevent wrong calculation results. Other related research works [4], [31] proposed Blockchain-based traffic event/message validation and trust verification frameworks to secure message dissemination and to tackle the problem of the incorrectness of data and its negative impact in vehicular environments. They present a proof-of-event (PoE) consensus concept. This consensus

concept aims at confirming event occurrences where vehicles periodically transmit their traffic information via CAM messages to RSUs which collect them and broadcast warning messages via DENMs once the event is valid. We believe that periodically sending messages occurs a huge overhead and delays vehicles' communications especially in congested environments. This is in addition to the Blockchain overhead that occurs while broadcasting/validating blocks and transactions over the entire network especially when the number of nodes is huge. In [4], beacons are used to exchange vehicle locations throughout the network and the PoW consensus is adopted. RSUs and a specific type of vehicles are considered as miners. These initiatives present several drawbacks because they do not consider the dynamic nature of vehicular networks and their high-performance requirements in critical situations. In other words, the huge resource consumption of the PoW consensus mechanism and its block generation and validation time do not meet VANET latency requirements especially when both RSUs and vehicles are miners [4]. On the other hand, Li *et al.* [55] proposed a hybrid trust management scheme to evaluate both data and mobile nodes trustworthiness in order to secure vehicular networks. The proposed scheme assumes that RSUs are always trustworthy and base their data analysis on this assumption which is very risky [11]. Traffic data is collected using all nodes in the network to be centrally analyzed without paying attention to the huge resource consumption occurred to complete this process [55], [56].

To the best of our knowledge, most of the proposed blockchain architectures for VANETs are very general [4], [7], [26]–[28] in their applicability of the Blockchain in the considered scenarios. None of them provide a complete Blockchain congestion control/prediction system that dynamically records vehicles trustworthiness while accurately predicting congestion occurrence and proactively managing network services based on the heterogenous users' QoS priorities.

B. DATA CONGESTION PROTOCOLS & ARCHITECTURES FOR VANETs

Congestion control protocols in VANETs are divided into two groups of solutions, namely open-loop and closed-loop solutions. They are also classified into four categories of protocols namely power-based [45], [58]–[60], CSMA/CA-based [61], [62], rate-based [45], [63], prioritizing-based and hybrid protocols [20], [45], [57], [64]. Hereafter, we review congestion control protocols that are closest to our work.

Taherkhani *et al.* [20] addressed the problem of data congestion at intersections and proposed a hybrid, centralized and localized data congestion control strategy based on the K-means algorithm using RSUs at intersections. On the other hand, existing open loop approaches that aim at predicting and avoiding congestion are mostly distributed and waste considerable time and resources in calculations [20], [33], [45]. In fact, Taherkhani *et al.* proposed an open-loop and distributed congestion control protocol that prioritizes safety and service messages based on the content of messages and state of the network and then dynamically and heuristically schedules messages in the control and service channel queues [20]. However, the taboo heuristic is continuously running on each vehicle to execute priority and scheduling tasks either in the presence or absence of congestion in the network. Zemmouri *et al.* [33], [45] also introduced a distributed open-loop congestion prediction protocol that allows each vehicle to estimate the vehicular density around itself and use this information to adapt beacons transmission parameters according to the current state of the network. Again, each vehicle should perform several calculations which drastically decrease the protocol performance in terms of the response latency, accuracy, efficiency and reliability. On the other hand, a centralized predictive road traffic management system was proposed in [65]. It estimates the future traffic intensities at different intersections based on a modified linear prediction algorithm and re-routes vehicles to reduce traffic congestion and the total journey time. In this work, each vehicle periodically sends its information to the RSU every 5 seconds. RSUs forward vehicles information to a centralized unit that predicts the future traffic flow once per minute. More in line with predictive congestion schemes, Wu *et al.* [25] proposed an aggregate parameter based on weights for congestion detection that aims at monitoring the network performance while considering four aspects which are the average delay, throughput, message delivery ratio and overhead ratio. In this work, it is unclear whether the algorithm is centralized or distributed.

In a nutshell, congestion prediction protocols completely ignore VANET's strict latency and reliability requirements while predicting congestion. We argue that those requirements demand ahead-of-time prediction results with latencies of milliseconds [15]. Furthermore, and far beyond performing calculations to adapt appropriate transmission parameters and control the congestion [33], details about the beacon dissemination protocol that allows vehicles to learn about the current network conditions are missing in [33], [45]. In fact, the periodic and blind exchange of beacons can not launch and would stop the operating of the proposed short-term prediction scheme especially in a highly dense environment. Moreover, no details are given in [65] regarding the operating of the proposed linear prediction algorithm nor the broadcast protocol used to disseminate messages in the considered vehicular network. This might drastically impact the system performance.

On the other hand, other research works focus on the negative impact of data congestion challenges that lead to the network infrastructure ossification problem [34], [38]. Various research works proposed NFV/SDN-based approaches to tackle this issue in order to facilitate the design and programmability of next-generation wireless networks [71], [72]. To do so, they addressed several aspects of NFV challenges among which is the VNF placement NP-hard problem. Several research works proposed heuristics and approximate solutions to optimize the placement of VNFs

along with their chaining order in SFCs while considering the huge network load and carrier-grade requirements of NFV applications [34], [39], [40], [88]. Nevertheless, the proposed strategies are mainly reactive threshold-based approaches [3] and need time to converge. Few research works have recently appeared in the literature in order to deal with this issue using ML algorithms [36], [37]. Kim *et al.* [43] introduced a learning model to predict VNF resource demands using SFC data. This learning model can be used to solve the optimal placement of VNFs. However, it is only applied to one SFC. Rahman *et al.* [42] proposed a proactive learning classification model to auto-scale VNFs while considering dynamic traffic changes. The selected features used to build the ML classifier are the time of the day and the measured traffic at specific times of the day. The classification output is the number of VNF instances required to serve future traffic flows while considering QoS requirements. This work does not give details regarding the obtained measurement of traffic that are function of the time of the day. The provided features are not enough to output an accurate number of VNFs depending on the dynamic network changes. The latter can impact the performance of the proposed model when complex real-time traffic patterns are introduced. Moreover, the approach used to output the number of VNF instances is very limited and the whole proposed model discards VNF placement requirements that satisfy carrier-grade requirements of NFV applications and that should further include the VNF chaining, scalability and adaptability [66].

On the other hand, none of the proposed predictive and proactive VNF placement strategies have considered the problem of the VNF vulnerability to security threats while predicting the placement of VNFs and their chaining order in SFCs. In fact, a single compromised VNF can entirely damage the whole network. Although the Blockchain technology has recently appeared as a powerful solution able to overcome current security challenges, only few research works have deployed Blockchain to tackle VNF challenges. In fact, Alvarenga *et al.* [67] proposed a Blockchain-based framework for secure configuration and migration of VNFs. Franco *et al.* [68] introduced BRAIN, an auditable solution that aims at discovering and selecting infrastructure providers able to efficiently host a VNF with regards to end-users demands. To the best of our knowledge, none of the previous research works provides proactive, fast, secure, and immutable prediction of VNF placement and their chaining order in SFCs while considering carrier-grade requirements of NFV and heterogenous users' QoS priorities.

III. PROBLEM STATEMENT

According to related research works, there exist three principal challenges that stay behind the lack of a consistent, flexible, dynamic, resilient and reliable congestion prediction system. More specifically, those challenges include 1) the entire negligence of security aspects while controlling congestion in vehicular environments, 2) the lack of congestion prediction protocols in vehicular environments and 3) the lack of secure, proactive and predictive VNF placement strategies that consider dynamic users' QoS priority requests and all carrier-grade requirements of NFV applications.

First, according to the regarded open and closed loop congestion control protocols and architectures, we argue that they inadvertently dismissed a fundamental and critical aspect of network and vehicular communications that is the huge vulnerability of RSUs and the whole vehicular environment to security threats [11], [84]. In fact, data information sent to RSUs and data disseminated among vehicles represent an easy target for malicious nodes [11]. Those nodes benefit from the anonymous nature of vehicular environments to easily create impersonation, privacy, jamming and forgery attacks [73]. Jamming attacks are very dangerous for all proposed approaches [2], [11]–[14], [73] especially in the case of centralized approaches such as the K-means congestion control strategy [20] and NFV-based frameworks [3], [36], [37], [42], [43]. It is considered as a severe Denial of Service (DoS) attack and can create fake data congestion. The latter can easily destruct the centralized K-means congestion control algorithm localized at RSUs [20] and inadequately consume a huge amount of computing resources, falsely allocated, to establish VNF-based orchestration policies to cope with fake congestion at the NFV orchestrator unit [3]. Hence, it is evident that trust among drivers and their vulnerability to security attacks, in harsh vehicular environments, constitute dangerous data congestion challenges and risks. They should not be neglected or taken for granted but rather be prioritized when devising solutions for data congestion issues.

To the best of our knowledge, SCOOL protocol and its extension called STEP [13], [14] is the only research work that took into consideration the dangerous consequences of security threats on congestion control protocols. Although this work isolates external attacks, it fails to defend against internal attacks such as forgery and DoS attacks. In fact, malicious nodes, in highly congested environments, take benefit of their proximity to propagate fake data or make a vehicle lie and deny receiving or forwarding certain packets, or broadcast large number of messages in a short period of time. This kind of attacks strongly and negatively impact intra-vehicle communication even with the presence of a powerful broadcasting protocol. Moreover, those attacks deteriorate the congestion control process centralized at RSUs.

On the other hand, SCOOL only focused on securing the vehicular environment but it failed to provide an efficient and secure broadcasting protocol to control the congestion efficiently and reliably. The absence of such a protocol leads to undesirable delays and exacerbates users' Quality of Experience (QoE). This gap complicates the process of building efficient routing tables in the network and ruins gateways' calculations of monitoring reports [13], [14]. The latter drastically impacts the accuracy of congestion control decisions taken at the RSU level and, more importantly, damages the whole network infrastructure, especially the performance of both SCOOL/STEP's secure efficient path recommendation protocol that is implemented at the RSUs.

Although those works consider security aspects to control congestion [13], [14], they are still in their initial stages [4]. In fact, SCOOL relies on central authorities to trace the real identity of traveling vehicles and remove malicious drivers from the network. However, central authorities revoke malicious nodes from the group only after the occurrence of an attack [13], [14] and this becomes harder and awful to detect in the case of DoS and forgery attacks in congested environments.

We argue that providing accurate traffic evaluation, by securing traffic and drivers' identity, constitute an initial and innovative step in the congestion control process. However, this step leaves the work incomplete and does not entirely solve the data congestion problem.

Other research initiatives [2], [4] tackled vehicles' trust management issues using the Blockchain technology to provide security and trust for message dissemination and traffic jam estimation. The new type of Blockchain proposed in [4] failed to meet VANET latency requirements. This is due to the lack of a reliable dissemination protocol and most importantly to its negligence of the huge PoW overhead and delay that occurred while generating new blocks at both centralized and distributed levels. On the other hand, the proposed Blockchain-based crowdsourcing model [2] incentivizes users to participate in the traffic prediction process through a neural network-based smart contract deployed onto the Blockchain network [22]. In this work, authors discarded several important aspects of the functioning of their proposed Blockchain model, and partially detailed the neural network model that predicts the probability of traffic jam. This makes the Blockchain considerations useless and worthless. In fact, the identity of PoA validators is not specified. Accounts are created for all vehicles willing to be part of the network without a prior selection and verification of those vehicles. This could easily insert false information into the smart contract neural network prediction process and deteriorate trust among all participating vehicles. The proposed crowdsourcing model only predicts traffic congestion at different locations and ignores the data congestion problem that locally occurs when vehicles share data among each other. All vehicles share their information with the smart contract deployed onto the Blockchain network without considering the huge resource consumption that this multiple sharing may occur. The smart contract events confirmation process is unknown. The complexity of the neural-network model is unknown, and the time complexity of the Blockchain validation consensus process is completely ignored. We believe that each validator is required to constantly run the neural network prediction process in order to add transactions to the Blockchain network. Reaching consensus in such situation is resources and time consuming, it ossifies the operating of the whole Blockchain network which does not go along with VANET's latency requirements.

Given the aforementioned challenges and gaps, we argue that in addition to securing vehicles' identity and ensuring accurate traffic, it is mandatory to fill the gap of predictive data congestion control mechanisms and devise, at the edge of the network, a secure, immutable and dynamic data congestion prediction mechanism. This mechanism should learn in real-time about congestion and automatically and reliably predict its occurrence, ahead of time. It should then determine and send appropriate network parameters to vehicles, at the most appropriate time, to smooth vehicles' data transmission. This would allow the network to alleviate undesirable delays and other congestion consequences and most importantly to ease the process of detecting and removing malicious nodes.

Second, existing congestion avoidance approaches are mostly distributed [20], [33], [45]. We argue that avoiding congestion in non-crowded areas represents a huge waste of network resources and requires efficient predictive strategies able to determine the appropriate time to avoid the happening of congestion. Constantly running avoidance strategies distributely on vehicles drastically impacts the performance in terms of the response latency and reliability metrics, that are of paramount importance in congested vehicular environments. Those metrics are essential to build and run several kinds of applications, among which are safety applications that require latencies of milliseconds to save lives and decrease accident rates. The few existing predictive approaches are incomplete [25], [33], [41], [45], because they do not provide enough details about the operating of their algorithms. They completely ignore the importance of underlying communication protocols based on which they compute metrics to predict/detect the congestion, and some of them do not tackle the data congestion problem [2], [41], [44]. We believe that an efficient and reliable communication protocol would decrease the execution frequency of the proposed congestion detection algorithms which can optimize the network resource consumption and satisfy users' QoS requirements.

Third, traffic applications and services range from safety and traffic efficiency applications to online social and mission critical applications [15], [35]. Safety applications require fast and reliable warning data transmission [35]. Online gaming applications and services are also very sensitive to delay, while location-aware video streaming services are more sensitive to message delivery [15]. Maintaining those QoS priorities in emergency situations like accidents and natural disasters represents a big challenge for broadband networks. In fact, the content of these applications is shared among vehicles and causes severe congestion at the servers' level. This congestion is due to the huge demand of periodic content update at the servers since the applications' content is highly correlated with the dynamic behavior of vehicles and their individual and various needs and QoS priorities on the road [72].

Today, service provisioning process relies principally on the deployment of middleboxes [8], that often form a service chain composed of chained network functions to certain traffic flows [8]. However, the performance of service chains is strictly hindered and constrained by the proprietary source code of heterogeneous hardware-based middleboxes which require manual features upgrade in hardware-based network

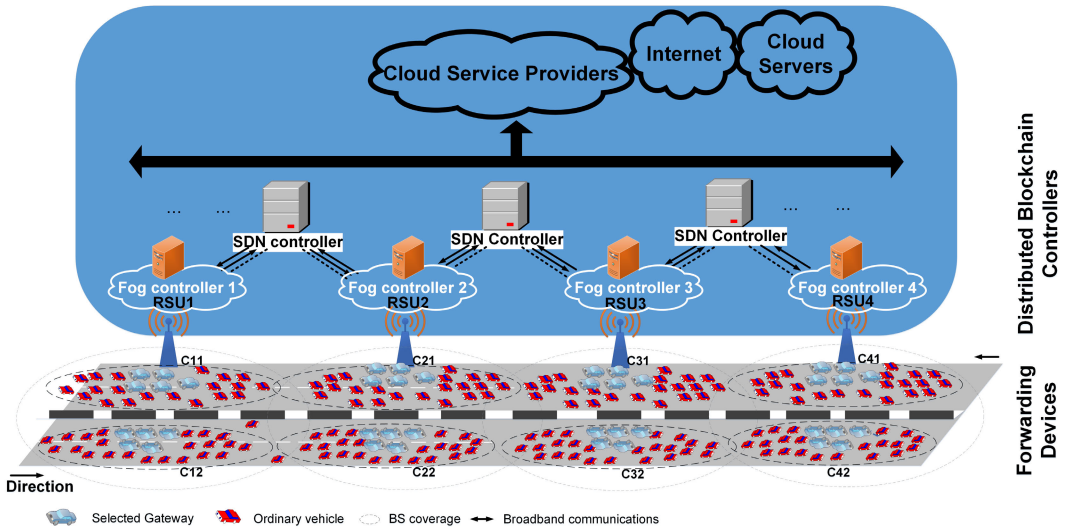

FIGURE 1. BCOOL topology.

functions to meet service chain requirements. The latter ossifies the network [38] and leads to high OPEX and CAPEX costs [8], [38]. NFV copes with this challenge through the virtualization of middleboxes-based network functions and deploys them on VMs [8]. This virtualization of network functions leads to great flexibility and programmability in the deployment of SFCs. However, this raises an important issue that relates to the placement and chaining of VNFs given the restricted and limited network resources and the huge and heterogeneous QoS priorities and requirements. Several research works [34], [39], [40] proposed approximate solutions and optimization heuristics to tackle this NP-hard problem [34]. However, those traditional solutions often require considerable time and resources to converge, which does not match the urgent and QoS priority requirements of user needs in terms of time-sensitive and critical applications. In addition, and more importantly, those solutions do not proactively predict the deployment of VNFs in SFCs neither pay attention to the vulnerability of VNFs to security attacks [36], [37]. Few research works have recently tackled the VNF resource prediction problem [36], [37], [42], [43]. However, they incorrectly use the real-time traffic information and some of them ignore the QoS priorities [90] which negatively impacts the accuracy of their prediction ML mechanisms. Furthermore, those works do not consider all carrier-grade requirements of NFV applications and failed to automatically and proactively provide fast accurate and secure VNF placement decisions in the context of SFCs. In fact, a compromised VNF can threaten the whole traffic that passes through SFCs. Therefore, it is of paramount importance to provide fast and accurate VNF placement decisions using a secure and proactive strategy that predicts the best and accurate placement of VNFs based on past placement decisions w.r.t highly dynamic users' QoS priorities.

IV. SOLVING METHODOLOGY & ARCHITECTURE

To the best of our knowledge, none of the previous networking architectures fully tackled the considered challenges. As a consequence, we introduce the first novel, flexible, dynamic, resilient, consistent and rich Blockchain COngestion contrOL (BCOOL) system that relies on the heterogeneity of promising networking paradigms -namely ML, SDN and NFV- to fill the gap of trustworthy Blockchain congestion prediction systems. BCOOL is a Blockchain-based distributed NFV-SDN architecture that brings the network and ML intelligence along with computing resources to the edge of the network using a distributed fog computing infrastructure. This greatly minimizes the processing and communication delay between vehicles and infrastructure resources and speeds up the PoA validation process that is executed by edge controllers. BCOOL creates and consolidates trust among distrustful controllers using on/offchain smart contract functionalities and allows them to reach consensus upon the execution of those contracts without requiring a third-party central authority. More specifically, BCOOL's modules aim at efficiently, dynamically and immutably managing vehicle and message trustworthiness, at the edge of the network, while providing fast, accurate and real-time prediction of congestion, ahead of time, to facilitate the proactive and automatic prediction of SFC deployment decisions that consider both dynamic users' QoS priority requests and all the carrier-grade requirements of NFV applications. It dynamically records vehicle, message trustworthiness, and prediction transactions in a distributed ledger and enforces the scalability and efficiency of its three main modules using novel hybrid on/off-chain smart contracts.

The first novel module is a hybrid Blockchain Fog-based Distributed Trust Contract Strategy (FDTCS) that efficiently, immutably, and dynamically manages vehicle and message

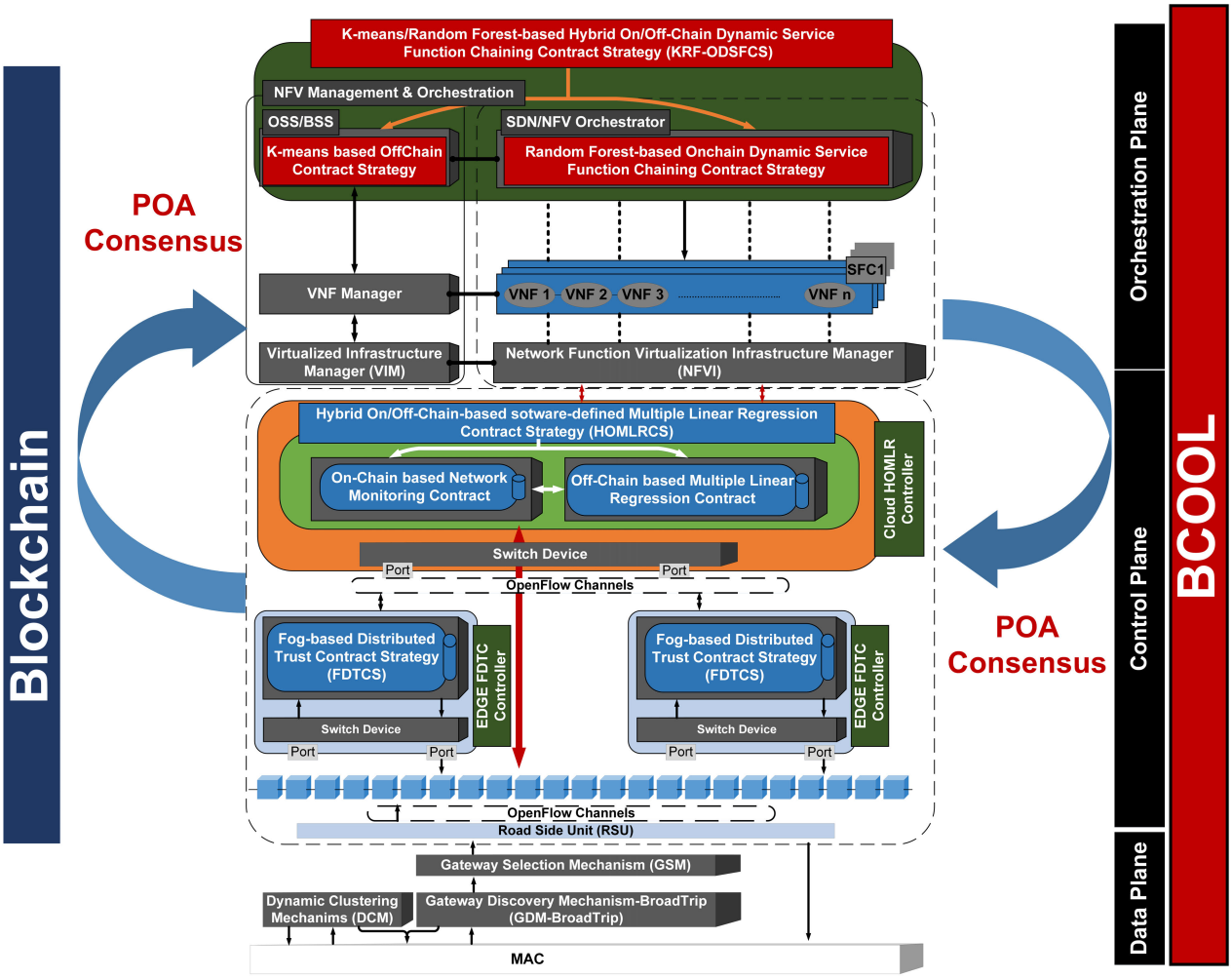

FIGURE 2. BCOOL architecture.

trustworthiness to prevent malicious nodes from acquiring access to the infrastructure.

The second novel module is a Hybrid On/Off-Chain-based software-defined Multiple Linear Regression Contract Strategy (HOMLRCS) that relies on the output of the first module and that aims at 1) continuously and efficiently learning and monitoring the network at the SDN controller, 2) predicting current level of data congestion and 3) determining and sending network parameters to vehicles, ahead of time, to alleviate the occurrence of congestion.

The third novel module is a K-means/Random Forest-based On/Off-Chain Dynamic Service Function Chaining Contract Strategy (KRF-ODSFCS) that relies on the output of former strategies and that aims at fastly, securely and proactively predicting SFC deployment decisions while considering dynamic users' QoS priority requests and all requirements of NFV applications.

A. ARCHITECTURE

Figure 2 portrays the BCOOL architecture that is mainly composed of three modules: The Hybrid Blockchain Fog-based Distributed Trust Contract Strategy (FDTCS), the Hybrid On/Off-Chain-based software-defined Multiple Linear Regression Contract Strategy (HOMLRCS) and the K-means/Random Forest-based On/Off-Chain Dynamic Service Function Chaining Contract Strategy (KRF-ODSFCS).

Hereafter, we describe the architecture's topology, our threat model considerations along with some assumptions.

1) TOPOLOGY

The topology of BCOOL is hybrid and is illustrated in Figure 1. The main network scenario is divided into three parts: the distributed vehicular network, the Blockchain-based distributed controllers' network and the network service providers. In the network scenario there are five kinds of nodes: ordinary vehicles, selected gateways, server nodes and SDN controllers that orchestrate FoG controllers and manage their resources. Vehicles are equipped with GPS devices allowing them to obtain their locations, speeds and directions.

2) ASSUMPTIONS

We assume that vehicles communicate with each other and the infrastructure, represented by RSUs, using the 5.9 GHz

Dedicated Short-Range Communication (DSRC) standard and Wireless Access in Vehicular Environment (WAVE). IEEE 1609.2 and IEEE 802.11p are two WAVE standards. We assume that security functions that include encryption key management, authentication, and so on are provided by IEEE 1609.2 and IEEE 802.11p standards. We also assume that controllers have appropriate and legal rights and are validators in our architecture. They are responsible for executing smart contracts, creating and approving contract transactions and adding blocks to the Blockchain network.

3) THREAT MODEL

In this paper, we suppose that there exist active adversaries that not only monitor the exchanged messages in the network but also execute spoofing actions. They introduce fake alerts, update data notifications and propagate lies that might induce undesirable effects. Moreover, adversaries can also disguise their identities to carry out malicious acts. In this work, we do not consider collusion attack, we leave it for future work.

B. SOLUTION METHODOLOGY

1) FOG-BASED DISTRIBUTED TRUST CONTRACT STRATEGY (FDTCS)

a: CONCEPT

FDTCS is a hybrid Blockchain trust contract strategy that aims at immutably, reliably and dynamically recording the message and recent vehicle trustworthiness in a distributed edge-based Blockchain network without requiring a third-party authority. FDTCS is the BCOOL's building block that promotes accurate and secure data congestion prediction. FDTCS prevents dishonest and malicious vehicles from misleading cooperative drivers and getting access to the network infrastructure.

b: ARCHITECTURE

FDTCS architecture elements are illustrated in Figures 1 and 2. This contract strategy is deployed at the fog controller level and only interacts with trustful and stable gateways to determine vehicles trustworthiness while saving bandwidth consumption and computing resources. It continuously updates message and vehicle trustworthiness and records history and recent vehicle trust scores in a chain of blocks, as long as PoA consensus is reached among fog controllers.

c: ALGORITHM

FDTCS operates as follows. Vehicles are first clustered using an efficient clustering strategy [6], [17]. To discover their neighborhood and the surrounding environment, vehicles in each cluster exchange notifications about the events they observe. Notifications are disseminated efficiently and reliably using a reliable dissemination protocol [16]. Vehicles proceed to the selection of a minimum number of gateways, upon the end of the discovery process, in order to get access to the fog controller while saving network resources. Those gateways are selected based on two metrics: links stability with cluster members and trust. The fog controller, on the other hand, keeps track of all notifications received and is in charge of computing and updating the General Trust Score (GTS) for all vehicles based on reports received from gateways. It then, broadcasts updated GTSs to clustered vehicles. It is worth noting that the vehicles' GTS is first provided by a central authority and then updated at the fog controller.

Gateways Selection: To proceed with gateways selection, vehicles compute Cluster Trust Scores (CTS) (explained below) for every vehicle from which they received the event's notification. They also check the stability (explained below) of their links and store this information in a routing table. Then, they select gateway vehicles with which they have strong stability (i.e., they drive with approximately the same velocity) and whose CTS and GTS for the observed event are high.

- Trust metric is expressed in terms of the veracity of notifications received from other vehicles. Each receiver computes a CTS based on its own experience of the received notification from each sender. For example, if a sender reports a notification for an event, the receiver observes and verifies the event and then judges the veracity of the sender's notification. If the sender's claim is judged honest, then the sender's trust is increased as follows:

$$cts_x^y(l+1) = cts_x^y(l) + m(1 - cts_x^y(l)), \quad m \in [0.1] \tag{1}$$

Otherwise, the sender's trust score is decreased as follows.

$$cts_x^y(l+1) = cts_x^y(l) - n(1 - cts_x^y(l)), \quad n \in [0.1] \tag{2}$$

where cts_x^y is the cluster trust score that vehicle x computed for vehicle y after having received a number of notifications from vehicle y. We use small values for m and n to prevent malicious nodes from gaining trust so quickly. The values of n are also small to alleviate the case of inaccurate judgment. The computed CTS is then added to the sender's GTS in order to obtain the trust score of each vehicle. CTS and GTS values are both between 0 and 1.

- Stability is used by each receiver to predict its link expiration time with the sender. The stability of links between the sender and receivers is defined by the Link Expiration Time (LET) and the Route expiration Time (RET) metrics. The LET defines the expiration time of the link between the sender and its next hop receiver. Whereas, the RET defines the expiration time of the link that expires first along a route composed of multiple (n-1) links. According to [54], if we consider two adjacent vehicles i and j with coordinates (x_i,y_i) and (x_j,y_j), a transmission range R, moving with speeds v_i and v_j in directions θ_i and θ_j, respectively, the estimated LET is

$$LET = \frac{-(ab+cd) + \sqrt{(a^2+c^2)R^2 - (ad-bc)^2}}{a^2+c^2} \tag{3}$$

where, $a = v_i \cos\theta_i - v_j \cos\theta_j$, $b = x_i - x_j$, $c = v_i \sin\theta_i - v_j \sin\theta_j$, $d = y_i - y_j$. The stability of a link is high $LET_{ij} = \infty$,

if and only if, both vehicles i and j move with approximately the same speed $v_i = v_j$ and in the same directions $\theta_i = \theta_j$. The RET consists of the link that expires first in a route and is computed as follows:

$$RET_{n-1} = \min\{LET_{j,j+1}\}, \quad j = 1 \ldots .n-1 \tag{4}$$

Fog-Based Vehicles' Trust Contract Evaluation: Once appropriate gateways are selected, they locally determine the trustworthiness of vehicles from which they received notifications regarding incidents. To do so, they compute, for each received event's notification, the sender's CTS and add it to the sender's GTS. Then, gateways send simultaneously their trust reports to the fog controller where the trust contract strategy is deployed (see Fig.2).

The reception of gateways' trust reports triggers the trust contract to compute and continuously update GTSs for each vehicle based on the updated received reports that are related to notifications issued by vehicles. Each sent report to the Fog controller contains notification messages that correspond to the report and their sender identification number. It also contains the gateways' trust evaluation of the received notification. The fog-based trust contract determines the GTS of a vehicle based on both the mean score of the vehicle's past GTS and the current vehicle's trust score $curr_{gts}$. The latter is computed based on the recent received updates for notifications issued by a vehicle since its last GTS calculation. More specifically, the GTS of a vehicle x can be written as follows:

$$gts(x) = \alpha mean_{p.gts}(x) + (1-\alpha) curr_{gts}(x), \alpha \in [0.1] \tag{5}$$

we choose to give more weight to the mean score of past GTS of a vehicle in a way that the general trust for a vehicle changes slightly. So, the mean general score of a vehicle weigh much more than the current score $curr_{gts}$. Therefore, the value of α is close to 1.

It is worth noting that we define a threshold for notifications related to an incident while computing $curr_{gts}$. The trust contract uses the gateways' attributed scores about the recent notification and analyzes the reports provided by the gateways. For each sender that is publishing a notification, the trust contract receives updated self-evaluations from all gateways who received the notification, observed and self-evaluated the event's notification update and were able to judge its truthfulness (see Algorithm 1, lines 2-4). More specifically, the $curr_{gts}$ score of a vehicle x is calculated as follows:

$$curr_{gts}(x) = \Sigma_g \frac{gts(g).b_g}{\sum_g gts(g)} \tag{6}$$

g belongs to the set of gateways (G) who received notifications issued by the vehicle x since its previous GTS was calculated.

$gts(g)$ is the current general score of g and b_g is a value that is either equal to 0 or 1 and that represents the g's self-verifications of x notification. We compute the $curr_{gts}$ using gateways' GTSs which are trustworthy vehicles in the considered cluster to have a great influence on the current score. The $curr_{gts}$ will be added to the mean score of past gts to obtain and update the general trust score of the vehicle x.

Algorithm 1 The Trust Contract

```
1: task Determining trustful nodes in the network
2:   upon True do
                          {The contract is waiting to be triggered}
3:     for all v ∈ V do
4:       GTS(v) ← α.mean_{p.gts(v)} + (1 − α).curr_{gts(v)}
         {Computes the GTS of vehicle v}
5:       if threshold ≤ GTS(v) ≤ 1 then
6:         Trust(v) ← True
7:       else
8:         Trust(v) ← False
```

Vehicles' trust scores (see Algorithm 1 line 5-8) and the corresponding notification messages are timestamped, validated using the PoA consensus mechanism and released to the Blockchain network. The Blockchain manages and stores the history of the vehicles' GTS and keeps track of vehicles' recent trustworthiness in an immutable and reliable manner which serves as a ground-truth for other network entities in the global network. Then, the GTS is broadcasted to vehicles through RSUs and gateways.

It is worth noting that the process of gateway selection is automatically retriggered once vehicles start losing connectivity with actual gateways. The latter greatly contributes to the vitality of the network.

d: COMPUTATIONAL COMPLEXITY

The complexity of our FDTC strategy is linear and includes the complexity of the dissemination protocol, namely BroadTrip used to exchange event notifications among vehicles, the complexity of the gateway selection mechanism and the complexity of the PoA consensus protocol.

The complexity of BroadTrip [16] is mainly evaluated upon the delivery of notifications at the MAC layer. In this layer, the notifications' retransmission is scheduled. For each received notification, we go through all n notifications received and check if they are ready for retransmission and if they are originated from opposite directions of the receiver. If this is the case, notifications are paired using network coding and retransmitted at the price of one notification. We then go through other notifications that are ready for retransmission and cannot be coded with other notifications to retransmit them. Therefore, the computational complexity of BroadTrip is $O(n)$.

Upon the delivery of all vehicle notifications, vehicles proceed to the selection of gateways. The complexity of the gateway selection mechanism is $O(v)$, where v is the number of vehicles in the cluster.

Then, upon the reception of gateway reports at the fog controller, the trust contract is triggered and updates the

GTS for each vehicle v based on the uploaded reports. This operation scales with $O(v)$.

This mechanism is followed by the PoA consensus protocol that operates on a set of C trusted authority controllers. Each controller is identified by a unique *id* and at least $C/2 + 1$ controllers run consensus to add blocks in the Blockchain network. This means that each authority controller is only allowed to broadcast a block every $C/2 + 1$ blocks. Therefore, at any time there are at most $C - (C/2+1)$ authority controllers allowed to broadcast a block [86]. Thus, the complexity of our consensus algorithm scales with $O(C)$ where C is the number of authority controllers. Since the number of vehicles is large compared to both the number of PoA consensus controllers and the number of notifications that vehicles could exchange, then the global computational complexity of FDTC strategy is $O(v)$.

2) HYBRID ON/OFF-CHAIN-BASED SOFTWARE-DEFINED MULTIPLE LINEAR REGRESSION CONTRACT STRATEGY (HOMLRCS)

a: CONCEPT

HOMLRC strategy aims at securely monitoring, fastly and accurately predicting the occurrence time of congestion, ahead of time, while relying on an efficient and reliable data dissemination protocol. HOMLR establishes trust and prevents data tampering once the execution result of the hybrid On/Off Chain contract is confirmed by Blockchain validators and added as a transaction to the Blockchain network.

b: ARCHITECTURE

HOMLRCS architecture is illustrated in Figures 1 and 2. In this strategy, vehicles exchange their messages using the Broadtrip protocol [16] and only communicate with honest vehicles defined by the fog-based trust contract. It is worth noting that vehicles' Broadtrip routing information is transferred and recorded as transactions on the Blockchain in order to stop malicious vehicles from generating routing transactions. The FoG controller continuously feeds the SDN controller with the received routing information of trusted vehicles.

The SDN controller monitors the network based on the received information, fastly and accurately predicts the occurrence of congestion using our HOMLR contract strategy. HOMLRCS predicts the occurrence of congestion ahead of time using the current network QoS metrics and based on past decisions. We split the functions of our smart contract, at the SDN controller, into an OnChain contract and an OffChain contract to mainly save computing resources during the mining process. In fact, smart contracts are mainly onchain and executed by miners. The Multiple Linear Regression (MLR) algorithm is computationally intensive [21] and would reduce the scalability of smart contracts, because miners could not generate any new block before the end of the execution of the smart contract by all miners. This might open the door to security attacks where adversary nodes skip the mining process (due to heavy MLR calculations) and add other new blocks to the Blockchain network.

To mitigate this issue, we outsource the execution of MLR to some honest participants in the context of an Off-chain contract. The Off-Chain MLR output execution will be then submitted to the onchain contract for verification and mining. The execution result of the hybrid On/Off-Chain smart contract cannot be tampered after being confirmed by the PoA consensus mechanism. Vehicles are then informed about the eventual occurrence of congestion ahead of time and can adapt their communication parameters to avoid its happening.

It is worth noting that contrary to other ML-based congestion detection mechanisms that rely on flooding protocol [13], [14], [20], our HOMLRC strategy relies on the Broad-Trip [16] protocol that optimizes the execution frequency of the HOMLR contract at the SDN controller and saves the additional MLR mining computing resources. Broadtrip schedules message retransmission using a smart and sophisticated combination of network coding [53] and location-based wait-and-count mechanisms [49]. The use of Broadtrip as an underlying communication protocol would allow predicting congestion ahead of time since the network's load and QoS metrics would be optimized as it is proved through simulations in Section VI.

In the following, we detail the operating of the MLR algorithm used in our strategy.

c: ALGORITHM

Generally, several network performance metrics determine the level of congestion in a vehicular network, among which are the packet loss, the throughput, the delay, the delivery rate and the network overhead. By monitoring change of these parameters, we can predict the level of congestion of the network and strategically alleviate the occurrence of congestion. However, finding correlation between those parameters is not trivial.

Our strategy is based on Multiple Linear Regression [21] which is a predictive analysis model able to estimate the level of congestion in real-time through an accurate continuous value, using a linear function of the network performance metrics. Unlike other regression algorithms, MLR investigates the relationship between dependent variable (or outcome) and several independent variables (predictors) [85]. It is a generalization of the simple linear regression model, suitable in non-linear real time problems, and is an error correction technique where learning is improved by training and experiences [46], [47]. In wireless networks, regression models can potentially be used to accurately predict network throughput, channel parameters, etc [46]. MLR is proven to outperform other regression algorithms for different reasons including extrapolation problem [85]. The general form of the MLR algorithm is:

$$f_\beta(x) = \Sigma_{i=0}^{k} \beta_i x_i \tag{7}$$

where $x_0 = 1$ is the intercept or bias feature, $f_\beta(x)$ is the first order model with k variables, β_i are the regression

coefficients, β_0 is a constant, x_i are the predictors (or independent variables) and k is the number of independent variables. The estimation of the regression coefficients can be done using the least squares technique. These coefficients are approximated such that the mean squared difference between the observed values and the predicted values is minimized. As a result, the prediction fits as much as possible the observed values. The Mean Squared Error (MSE) function is as follows:

$$E(\beta) = \frac{1}{2n}\sum_{i=1}^{n}(f_\beta(x^{(i)}) - y^{(i)})^2 \tag{8}$$

where n is the size of the training set and $y^{(i)}$ is the i^{th} output in the labeled set. Our main optimization objective is to find the best regression coefficients that minimize the function $E(\beta)$. The gradient descent algorithm is used to compute the optimum β. where

$$\frac{\delta E(\beta)}{\delta \beta_j} = \frac{1}{n}\Sigma_{i=1}^{n}(f_\beta(x^{(i)}) - y^{(i)})x_j^i \tag{9}$$

λ is the learning rate and β_j the j^{th} parameter in the vector of coefficients β. The convergence of the gradient decent algorithm can be accelerated when independent variables (predictors) are in the same dimension. Predictors scaling technique allows the algorithm to reach the optimum faster. The predictor scaling equation is given below:

$$x_i = \frac{x_i - \rho_i}{s_i}, \quad i = 1, \ldots, k \tag{10}$$

k is the number of predictors, x_i the predictor i in the training set, ρ_i the mean value of the predictor x_i and s_i the standard deviation of the predictor x_i. The convergence of the gradient descent can also be accelerated by modifying the learning rate λ. If λ is very small, the algorithm will take time to converge, and if it is very large, the algorithm will diverge. As a consequence, we must try different values of λ to reach convergence.

HOMLRCS Algorithm and Variables Definition: Hereafter, we detail the operating of our approach and define variables used in our MLR function. As illustrated in the flowchart of our algorithm (see Figure 3), the whole contract is split into two contracts. The light functions of our contract are executed onto the Blockchain using an on-chain contract, while we leave our heavy MLR algorithm to an anonymous off-chain contract. Our on-chain contract verifies the rate at which vehicles send their own messages and is publicly executed by all miners. If the message rate increases and reaches a predefined threshold, then the off-chain contract is invoked and proceeds to the anonymous computation of the current network variables and the prediction of the congestion level using gradient descent algorithm based on our selected independent variables also called predictors.

The throughput, the delay, the message delivery and the message overhead are considered key QoS metrics indicators of congestion in vehicular networks [20]. They are the predictors of our MLR function which can be written as

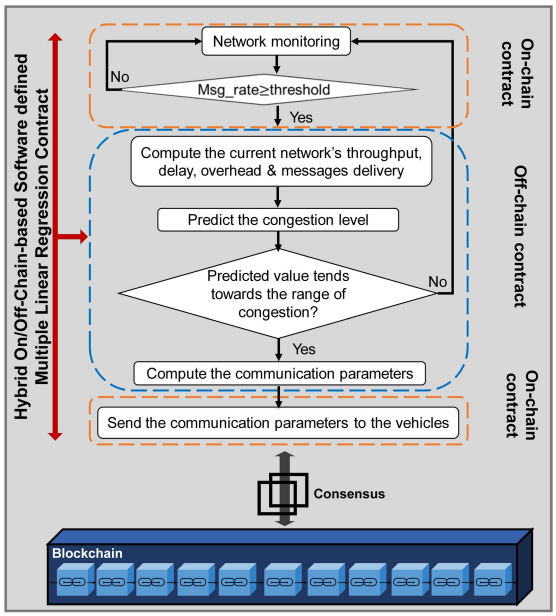

FIGURE 3. HOMLR flowchart.

Algorithm 2 Gradient Descent

```
proc GD(D, β^0)
    β ← β^0
    while not converged do
        for 0 ≤ j ≤ k do
            β_j ← β_j − λ.(1/n) Σ_{i=1}^{n} (f_β(x^(i)) − y^(i)) x_j^(i)
            {λ is the learning rate, n is the training data size and k is the number of predictors}
    return β
```

follows:

$$f_\beta(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 \tag{11}$$

where x_1 is the throughput, x_2 the delay, x_3 the message delivery and x_4 the message overhead.

β_0, β_1, β_2, β_3 and β_4 are the parameters of the function.

x is the vector (x_1, x_2, x_3, x_4).

The cost function is calculated using equation (8), where:

$E(\beta)$ is the MSE cost function, $f_\beta(x^{(i)})$ the estimated value of network congestion when the throughput, the delay, the message delivery and the message overhead are represented by $x^{(i)}$.

$y^{(i)}$ is the network congestion value in the training set.

$x^{(i)}$ is the vector containing the four predictors, namely the throughput, the delay, the message delivery and the message overhead in the training set. n is the size of the training set. The gradient descent algorithm (see algorithm 2) finds the optimum β which minimizes the cost function $E(\beta)$ while varying the learning rate λ for each iteration. β_0, β_1, β_2, β_3 and β_4 are the regression coefficients of the cost function to be optimized, and k is the number of features that is equal to 4.

Once the optimal values of β are found, the off-chain evaluates the regression function output and determines the value and the current congestion level along with the appropriate communication parameters, namely the transmission range and rate and the contention window (CW_{min}) to be used by vehicles to alleviate the occurrence of congestion.

The off-chain submits the algorithm computation result to the on-chain contract which sends a contract creation transaction to the Blockchain network. Upon miners receive the transaction, they execute the contract code to verify the correctness of the code execution which consists of checking whether the predicted value of congestion matches with the computed result. If the transaction is validated, it is included into a new block and chained into the Blockchain after being confirmed by the whole network using the PoA consensus.

Then, the FoG controller communicates the determined communication parameters to vehicles to alleviate the possible occurrence of congestion and smooth data transmission.

d: COMPUTATIONAL COMPLEXITY

The computational complexity of our HOMLRC strategy includes the complexity of the BroadTrip protocol.

As discussed in Section B.1, the computational complexity of BroadTrip is $O(n)$. If the message rate increases and reaches the predefined threshold, we run HOMLRC strategy at the SDN controller level that consists of executing the least square algorithm to find optimal values of the regression coefficients β_i that minimize the MSE. The MSE function can be written in matrix terms as follows

$$\beta.XX' = YX' \tag{12}$$

Which is equivalent to

$$(XX')^{-1}\beta.XX' = (XX')^{-1}.YX' \tag{13}$$

And since

$$(XX')^{-1}.XX' = 1 \tag{14}$$

We have

$$\beta = YX'.(XX')^{-1} \tag{15}$$

For a least square regression with N training examples and K features, it takes $O(K^2N)$ to multiply X' by X, $O(KN)$ to multiply Y by X' and $O(K^3)$ to compute the inversion of $(XX')^{-1}$ and use that to compute the matrix product of $YX'.(XX')^{-1}$ [87].

Since $O(N)$ asymptotically dominates $O(KN)$ we can neglect the $O(KN)$ part. Again, since $N>K$, then $O(K^3)$ is asymptotically negligible in front of $O(K^2N)$. Therefore, the computational complexity of HOMLRC strategy is $O(K^2N)$. Since the number of messages that vehicles could exchange is large compared to the number of PoA consensus controllers and the size of the training set, then the global computational complexity of HOMLRC strategy is $O(n)$.

3) K-MEANS/RANDOM FOREST-BASED ON/OFF-CHAIN DYNAMIC SERVICE FUNCTION CHAINING CONTRACT STRATEGY (KRF-ODSFCS)

a: CONCEPT

KRF-ODSFC contract strategy aims at providing a dynamic, secure, proactive, fast and accurate prediction of VNF placements and their chaining order decisions in the context of SFCs based on recent placements and while considering online QoS priority classes of SFC user requests along with carrier-grade requirements of NFV applications.

We consider SFCs based on network services of the virtual Evolved Packet Core (vEPC) framework [89]. vEPC encompasses four main VNFs, namely the Mobility Management Entity (MME), the Home Subscriber Service (HSS), the Serving Gateway (SGW) and the Packet Data Network Gateway (PGW). It is worth noting that our strategy can be applied to any type of SFC.

Our proposed strategy is a hybrid On/Off-Chain smart contract that represents a hybrid and joined execution of the Off-Chain K-means clustering contract along with the On-Chain-based Random Forest prediction contract policy. We choose to split the functions of our contract strategy and outsource some of them off-chain to optimize the network resources and accelerate the whole mining process. We assume that offchain functions are executed by honest participants and that network service providers are not malicious.

The Off-Chain K-means clustering contract strategy aims at clustering user requests into various QoS priority classes based on the unlabeled traffic received from the SDN controllers using the HOMLRCS. Compared to other clustering algorithms, K-means is an efficient, fast and simple learning algorithm capable of processing large data. The major drawback of K-means algorithm is that it does not converge rapidly especially for big datasets [19], [76].

The Random Forest (RF) [1] is an ensemble decision tree algorithm that constructs several distributed trees, trained on various parts of the same training dataset, where the target (dependent) variable is known and where each tree outputs a response to build a classification or prediction model that predicts future responses.

In our strategy, we execute the training process offline, and the digest of the training data is uploaded to the OnChain Random Forest strategy. Moreover, the testing process consists of submitting the online set of the obtained K-means clusters of current QoS priority classes, received from SDN controllers, to the Onchain RF Dynamic SFC policy. The latter interprets these clusters by creating real-time if-then-else decision rules to efficiently and accurately predict the placement and priority-based chaining order of VNFs based on trained sample sets.

Service deployment is triggered by the publication of onchain RF contract transaction on the Blockchain network.

In the following, we detail the operating of our strategy. We first present its architecture and then describe its algorithm.

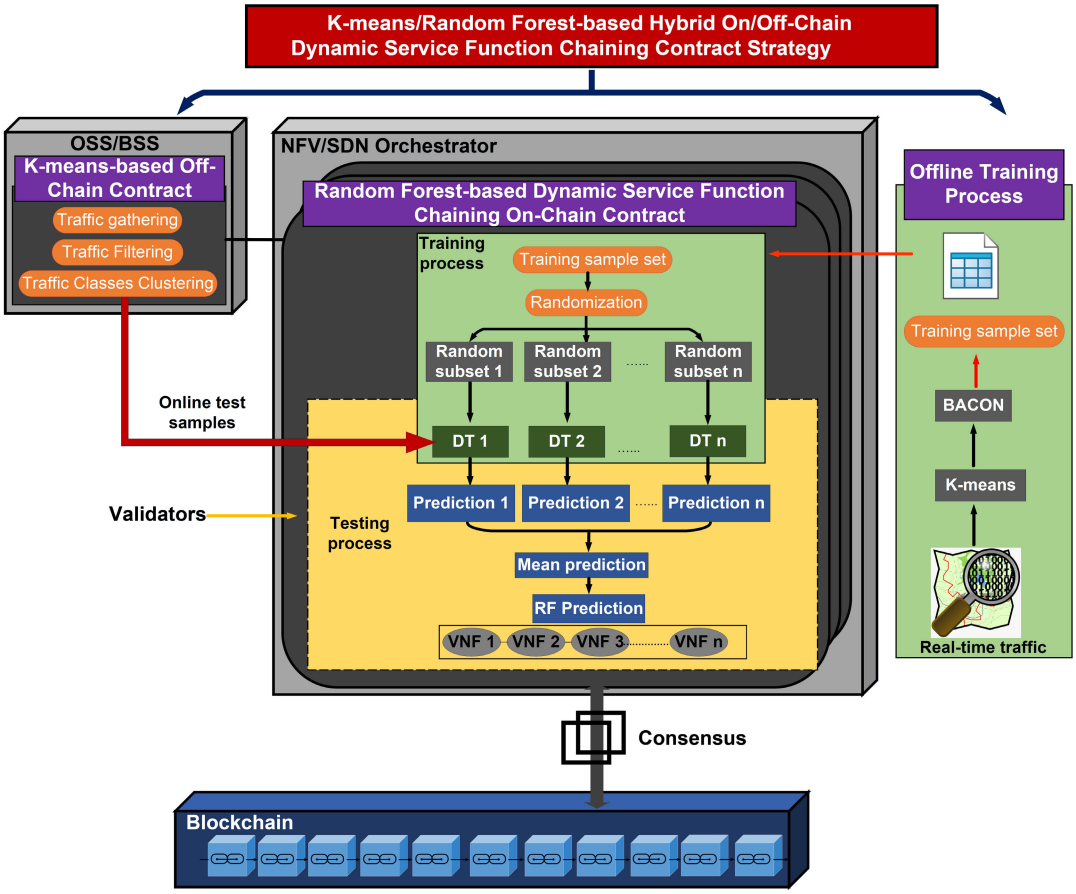

FIGURE 4. KRF-ODSFC block diagram.

b: ARCHITECTURE

Figure 2 illustrates the proposed Blockchain strategy in the ETSI NFV reference architecture. Our Offchain K-means contract can be part of the Operations Support System (OSS) and Business Support System (BSS) functions. OSS/BSS automatically assists the NFV-MANO element in the execution of network policies. This corresponds to the function of our Offchain K-means contract that builds clusters of QoS priority classes in real-time, based on SDN controllers input, and submits them to our RF On-chain Dynamic SFC contract policy that is part of the NFV/SDN Orchestrator. According to the ETSI definition [8], the NFVO orchestrates network services simultaneously with the NFVI and VNF Manager. It determines adequate policies per identified NFV application and links together appropriate and different types of VNF according to the QoS priority classes. Similarly, our RF OnChain DSFC contract policy outputs a prediction of VNF placements and their chaining order decisions, based on past decisions, w.r.t. various and current QoS priority classes clustered using K-means off-chain contract.

Those decisions are in the form of accurate predictions of optimal priority-based chaining and placement of appropriate VNFs that serve the various QoS priority classes of user requests while satisfying carrier-grade requirements of NFV applications. Those decisions constitute resource coordinated Dynamic SFCs where VNF types are shared among SFCs based on their functional dependencies and are managed by both VNF manager and NFVI resources.

c: ALGORITHM

Hereinabove, the block diagram of our KRF-ODSFC strategy which illustrates its operating. As shown in Figure 4, KRF-ODSFC principally relies on a hybrid combination of unsupervised and supervised algorithms, namely K-means and Random Forest (RF). Hereafter, we detail their operating in our strategy.

Offline Training Process: In this process, we first use K-means algorithm to cluster the users' traffic requests into different QoS classes. The QoS class defines the level of delay sensitivity of a traffic class. The K-means features are the connection duration, message length per connection and the variability of the intermessage arrival. We initially focus on four broad QoS classes regularly found in corporate networks that we list below from highest priority QoS class to lowest.

- The streaming class represents real-time traffic class which is the highest priority traffic and which requires low end-to-end delay. It is exemplified by RTP/UDP protocol and is

characterized by low to medium average message size and low connection duration.
- The interactive class is a low end-to-end traffic class represented by Telnet, RDP and FTP control. Traffic requests in this class have small message size but long connection duration.
- The transactional class specifies the high priority traffic class, is represented by DNS protocol and Oracle transactions and is characterized by small average message size and short connections.
- The bulk data transfer class specifies a low priority traffic class and requires high throughput for bulk data transfer without any real-time constraints. It includes applications such as FTP and is characterized by large or medium average message size and low to medium connection duration.

We then use a modified version of the BACON VNF placement heuristic [34] to generate the dataset labels that depicts the best placement of VNFs and their chaining order within SFCs. Our priority-based BACON heuristic aims at serving various requests, represented by QoS priority classes, by selecting the best placement for the VNFs that minimizes the end to end SFC delay and that considers the priority of the SFCs, which depends on the users' QoS priority classes, while sharing same VNFs and satisfying the various carrier-grade requirements of the NFV. The parameters of the BACON network include server resources, VNF computational resources, communication delay tolerance between VNFs, the communication delay between servers, the delay between two VNFs and the priority of the service requests at the traversed VNFs. The training process is conducted for each network at different times of the day for a certain period that may vary from days to a week.

The output of the offline training process, that is the near optimal VNF placement given the considered network conditions is then submitted to the On-Chain RF contract strategy.

The complexity of the training process includes the complexity of both K-means algorithm and BACON heuristic. The complexity of K-means relies on the amount of traffic T (user requests), the number of clusters K and number of iterations I needed for cluster convergence. It is equivalent to $O(kTI)$. In the worst case scenario, when the amount of traffic increases, the number of iterations increases systematically. Hence, the complexity of K-means in the worst case is $O(T^2)$.

On the other hand, the complexity of the VNF placement heurisitc is $O((s^3 - s^2)/2)$, where s is the number of available servers in the network. Overall, comparing the two algorithm complexities, we can see that K-means operates at a lower complexity.

OSS/BSS K-Means-Based Off-Chain Contract: The main objective of the K-means algorithm is to build clusters of QoS priority classes such that the similarities among QoS class members within the same cluster are maximal, while similarities among QoS classes from different clusters are minimal [19], [74]–[77]. K-means clusters a set of data into k number of clusters based on data features. The k number of clusters are represented by their centroids. For each data, K-means computes the Euclidean distance to all centroids and selects the minimum distance. The data belongs to the closest cluster where the distance between the data and the centroid is minimal. Then, the new centroid is computed for each cluster based on the mean coordinate of all members of each cluster. Finally, all the data are clustered based on the new centroid. K-means repeats these steps until the data cluster stabilizes [74]–[77].

More specifically, K-means mainly consists of the following steps: 1) selecting initial centroids of the K clusters; 2) computing the distance of each data to the centroids using squared Euclidian formula; 3) computing the new cluster centroids and finding closest centroids then repeat steps 2 and 3 until the members of the clusters no longer move.

In our strategy, the initial centroids for k clusters are the first k QoS classes of traffic received from the SDN controller. Features, number of clusters and number of iterations are three basic inputs of K-means clustering algorithm. Features are very important for the performance of K-means and should be specifically determined according to each problem [19], [75], [76]. The features of our K-means Off-Chain contract strategy are the connection duration, the mean of the message length per connection and the variability of the intermessage arrival. The number of clusters is the second input for K-means algorithm. The best number of clusters can be determined by running the clustering algorithm for different number of clusters [19], [75], [76]. The convergence of K-means algorithm is achieved when there are no changes in the members of the clusters. However, if it is not reached, the algorithm should be halted after a specific number of iterations. Algorithm 3 represents our proposed K-means Off-Chain Contract Strategy.

Upon the arrival of users' traffic requests, the K-means OffChain proceeds to building clusters of QoS priority classes for each instance and submits its computing results that consist of an instance of QoS priority classes to our OnChain RF contract strategy. This result serves as a test sample to the OnChain Random Forest contract strategy.

The testing process based on the RF algorithm aims at predicting the best placement and chaining order of VNFs for each new k-means test sample based on the generated training dataset which maps features (QoS priority of the service request, server resources, VNF computational resources, communication delay tolerance between VNFs, communication delay between servers, the delay between two VNFs) with labels (VNF placements and their chaining order).

RF-Based DSFC On-Chain Contract: The RF algorithm is the most robust stable and effective method in prediction [78]. It provides good performance compared to other regressors, including neural nets and trees [78]. It relies on bagging or bootstrap aggregating technique. Only a subset of sample from the original training set S are used to create individual trees, where $S^{(i)}$ is the i^{th} bootstrap. We then use a modified decision tree learning algorithm that selects, at each split in the learning process, a random subset of the features f of F

Algorithm 3 K-means Off-Chain Contract Algorithm

Input:
  T *{Traffic to be clustered}*
  k *{number of clusters}*
  $MaxIters$ *{limit of iterations}*
Output:
  $C = \{c_1, \ldots .c_k\}$ *{set of k clusters}*
  $L = \{l(t) \mid t = 1, \ldots .n\}$ *{set of cluster labels of traffic classes}*
$X = \{x_1, \ldots .x_n\}$ *{Set of n data points}*
$C = \{c_1, \ldots .c_k\}$ *{Set of k centroids}*
T is represented by X
  task Determining the inital centroids of the clusters
    $C = \{x_1, \ldots .x_k\}$ *{considering the first k classes of traffic as the initial centroids of the clusters}*
    for $1 \leq i \leq n$ **do**
      for $1 \leq j \leq k$ **do**
        $dist(x_i, c_j) \leftarrow \sum_{i=1}^{n} \sum_{j=1}^{k} \parallel x_i - c_j \parallel^2$ *{Euclidian distance of data point x_i to the centroid c_j}*
        $Determine(c_j, x_i)$ *{determining closest centroid c_j to x_i}*
        $l(x_i) \leftarrow \arg\min(x_i, c_j)$
  task Main loop of clustering
    $changed \leftarrow false$
    $iter \leftarrow 0$
    task Recalculate and update centroids, determining new c_i in C
      repeat
        for $1 \leq i \leq k$ **do**
          $c_i' \leftarrow \frac{\sum_{j=1}^{n} 1\{c_j = i\} x_j}{\sum_{j=1}^{n} 1\{c_j = i\}}$ *{calculate mean coordinate among all member of c_i}*
          $Determine(c_i', x_j)$
          $c_i' \leftarrow x_j$
        for $1 \leq i \leq n$ **do**
          for $1 \leq j \leq k$ **do**
            $dist(x_i, c_j) \leftarrow \sum_{j=1}^{k} \sum_{i=1}^{n} \parallel x_i - c_j \parallel^2$ *{Euclidian distance of data point x_i to the centroid c_j}*
          $Determine(c_j, x_i)$
          $minDist \leftarrow \arg\min(x_i, c_j)$
          if $minDist \neq l(x_i)$ **then**
            $l(x_i) \leftarrow minDist$
            $changed \leftarrow true$
        $iter++$
      until $changed = true \wedge iter \leq MaxIters$
  return (C, L)

to alleviate the correlation among single trees and to increase the prediction accuracy [79].

In general, two thirds of each training set are sampled each time a bootstrap sample is taken. The one third of remaining dataset is used for testing the prediction error of random forests [78].

Algorithm 4 RF On-Chain Contract Algorithm

Input:
  $S = \{(x_1, y_1), (x_2, y_2) \ldots .(x_n, y_n)\}$ *{A training set}*
  x_i *{input vector}*
  y_i *{Target response - chain of VNFs}*
  F *{Features}*
  B *{number of trees in forest B}*
Output:
  A prediction decision from all single trees
function RandomForest(S,F)
  $H \leftarrow \emptyset$
  for $i \in (1, \ldots .B)$ **do**
    $S^i \leftarrow$ *A bootstrap sample from S*
    $f \leftarrow$ *very small subset of F* *{at each node of tree h_i, select f features randomly}*
    Split on best feature in f
    $h_i \leftarrow RandomizedTreeLearn(S^i, F)$
  $H \leftarrow H \cup h_i$
  return $\hat{y} = \frac{1}{H} \sum_{h=1}^{H} p_h(x')$

It is worth noting that the training process is similar to the testing process. When the new K-means test sample is fed into the root of each decision tree, it is classified into either the right or the left child node until it reaches the leaf node. For each decision tree the prediction for future VNF placements can be obtained by calculating the mean prediction of all individual prediction trees. Algorithm 4 shows the pseudocode of our RF Onchain contract strategy.

Always all VNF placement predictions for onchain contracts are sent as a contract creation transaction to the Blockchain network. Upon receiving the transaction, miners include the contract output into the new block and include it in the Blockchain network after a successful verification of the onchain contract. An onchain address is a unique ID identifying the onchain contract and its information. The infrastructure provider address and the Onchain address form a hash format generated from a cryptographic key representing one address in the Blockchain. Thus, the onchain VNF placement prediction for each SFC user request as well as functions inside them are all securely recorded inside the Blockchain for decisions related to future audition. In fact, all miners can execute the onchain with the same input and get same outputs.

d: COMPUTATIONAL COMPLEXITY

The complexity of our KRF-ODSFCC strategy includes both K-means and Random Forest computational complexities.

In fact, the complexity of our OSS/BSS K-means-based Off-Chain Contract is linear and is equal to $O(kTI)$ where k is the number of clusters, T is the traffic classes and I is the number of iterations needed to complete the clustering process. We determine the number of iterations needed for the clustering process offline during the training process and we use the determined iteration values while online

clustering user requests into different QoS priority classes at the OSS/BSS level. Therefore, the main operation executed by our OSS/BSS K-means-based Off-Chain Contract strategy, while iterating using the predetermined number of iterations, is computing distances between data points and cluster centroids until the convergence is reached. This leads to a linear complexity in function of traffic classes $O(T)$.

The complexity of our RF-based DSFC On-Chain Contract strategy consists in the complexity of three phases; The first phase corresponds to the construction of the random forest, the second phase deals with the random selection of features at each node of the DT and the third phase deals with the execution of the test.

The complexity of the first phase depends on the complexity for building a complete unpruned decision tree that takes $O(n\log(n))$ where n is the number of records in the training set. When building random forests with k trees and for a subset of the features f of F sampled at each node, the complexity of the first phase would be $O(kfn\log(n))$. The complexity of the second phase deals with the randomization processes and is equal to $O(kn\log(n))$, while the complexity of executing a test is $O(k\log(n))$ [93]. During runtime, our RF-based strategy operates at a lower complexity which obviously means that its initial training phase would have already been completed and it would only executes testing requests. Therefore, the complexity of our KRF-ODSFCC strategy that includes the PoA consensus process is asymptotically dominated by the complexity of the K-means strategy that is $O(T)$.

Globally, when comparing BCOOL's modules complexities, we can evidently notice that during runtime, the HOMLRCS complexity $O(n)$ asymptotically dominates the FDTC and KRF-ODSFCC complexities because the amount of messages n is far greater than both the number of vehicles v and the amount of traffic T.

C. BCOOL FAILURE MODEL

BCOOL relies on the PoA consensus model which belongs to the family of Byzantine Fault Tolerant (BFT) protocols [86]. This BFT consensus protocol withstands Byzantine failures using collective validator decisions issued from working and non-working SDN controllers to reduce the impact of non-working SDN controllers. PoA was originally integrated in the Ethereum version called parity and Geth each of which offers PoA consensus protocols called Clique and Aura. Contrary to BFT protocols that allow a maximum of $F < N/3$ Byzantine nodes, where Byzantine means validators that are non-working or malicious, Aura and Clique withstand a maximum of $F < N/2$ Byzantine nodes. Generally, consensus in distributed environments requires the fulfillment of two main correctness requirements which are consistency and availability. Those requirements state that all SDN controllers should properly execute user requests in the same order and at approximately the same time. Angelis *et al.* [94] has recently proved that the Clique consensus protocol has the lowest message latency and that it beats both Aura and BFT message latencies. They also proved that BFT ensures strong consistency at the price of availability and that Clique is faster than BFT. According to their study, Clique provides availability and guarantees an acceptable consistency thanks to its GHOST protocol, whereas Aura guarantees availability with no consistency. Consequently, we adopt the PoA Clique protocol to withstand failures in predicting future responses at both our HOMLRCS and KRF-ODSFCS modules. The PoA Clique protocol withstands up to $F < N/2$ Byzantine validators and if those validators provide incorrect prediction information or fail to predict future responses they are voted out once the majority vote threshold is reached.

Additionally, the functions of our HOMLRC and KRF-ODSFC strategies are split into OnChain and Offchain contracts to mainly save computing resources during the mining process. The execution of the MLR prediction and K-means algorithms is outsourced to some honest participants in the context of Offchain contracts. If there is a disruption among those participants about the validity of the OffChain execution, then we execute the MLR prediction and/or K-means algorithms at the OnChain level in order to check the correctness of the Offchain participant results and to detect and redress the misbehavior of dishonest participants which caused the disruption at the Offchain level.

V. BCOOL ATTACK MITIGATION STRATEGIES

In this section, we demonstrate through effective strategies how BCOOL mitigates potential security attacks and provides resilient, secure, and trustworthy congestion prediction decisions in a safe vehicular environment.

A. RESISTANCE TO INTEGRITY ATTACKS

All transactions are signed and join other transactions in a block after being validated by more than half of authenticated servers using the PoA consensus mechanism. Transactions are hashed using a Merkle hash that is included in the block header and is called the Merkle root [80], [81]. The latter represents the hash of all transaction hashes that exist in a block of the Blockchain. A malicious vehicle would need to change the hash code of every block in the Blockchain network which is awful and requires huge computing resources. Consequently, the information in blocks cannot be tampered. Therefore, the network congestion and the SFC deployment decisions can be proactively, securely and accurately predicted ahead of time during the network lifetime.

B. RESISTANCE TO AVAILABILITY ATTACKS

The Blockchain network is based on the PoA consensus mechanism which is qualified for its efficiency compared to other consensus mechanisms, namely the PoW and PoS models. Using the PoA mechanism, the validation of only more than half of authenticated servers allows the storing of the transaction and the update of the ledger. Therefore, the system

generates blocks with high throughput which promotes the availability of services and data.

C. RESISTANCE TO CONSISTENCY ATTACKS

BCOOL records both history and recent message and vehicle trustworthiness in the distributed ledger which serves as a ground truth for other vehicles and network entities. BCOOL relies on the PoA consensus to approve and store transactions in blocks. Moreover, we split the functions of our smart contracts into onchain and offchain contracts to lighten the mining process and we also use an efficient and reliable data dissemination protocol to reduce the execution frequency of the off-chain congestion prediction contract strategy and further optimize the transaction validation process. The routing information is also stored on the Blockchain. Therefore, we state that the data stored in transactions signed with digital signatures in the Blockchain cannot be changed and that BCOOL supports both availability and strong consistency. This means that the public approval of transactions through the PoA consensus-based agreement guarantees the fact that all vehicles have the same Blockchain at the same time and that any request should wait until the approval of the current update. Hence, BCOOL also prevents the double spending attack.

D. RESISTANCE TO DDoS ATTACKS

In this attack, vehicles attempt to flood a number of RSUs by sending a series of transactions without any gap between periodic transmissions in order to interrupt their normal functioning. BCOOL overcomes this attack thanks to the decentralized maintenance of the Blockchain that guarantees the continuity of blocks' generation and validation even if some Blockchain validators go offline. In fact, BCOOL consensus mechanism exclude unavailable nodes from the list of validating nodes. Moreover, BCOOL consensus mechanism grants the block generation and validation tasks to only nodes able to defend against DDoS attacks.

E. RESISTANCE TO IMPERSONATION ATTACKS

BCOOL prevents impersonation attacks thanks to its strong anonymity that ensures both pseudonymity and unlinkability. Pseudonymity consists at providing pseudo-identity to protect user identities whereas unlinkability withstands the ability of malicious adversaries to derive user identities through de-anonymization inference attacks.

F. RESISTANCE TO 51% ATTACKS

In this case and in the context of PoA consensus, an attacker controls more than 51% of network nodes which is different from the PoW consensus 51% attack. In fact, an attacker in the PoW consensus takes control over the majority 51% of the network computational power resources. We believe that taking control of the computational power is much easier than controlling nodes. In fact, an attacker can boost the computational power to dominate the controlled network and this increases the percentage attack. In contrast to the PoW consensus, the PoA consensus does not rely on computational power to generate and validate new blocks. This makes PoA more robust than PoW.

VI. PERFORMANCE EVALUATION

Having described the details of BCOOL modules and analyzed their respective complexities, attack mitigation strategies and resilience to failures, we present in this section the performance evaluation of BCOOL. We aim at evaluating BCOOL's whole mining process performance through the measurement of the execution frequency of its contract components. To do so, we profoundly analyze its underlying communication protocol that fully dominates BCOOL computational complexity performance and mainly rules the functioning of its edge and cloud On/Offchain smart contract modules. Accordingly, we assess the performance of the HOMLRC-BroadTrip module that plays a key role for the performance of BCOOL at different levels. It monitors users' information exchange to accurately and securely predict the occurrence of congestion, ahead of time, at the SDN controller level, and it fully contributes to the provision of smooth real-time traffic flow to BCOOL's upper KRF-ODSFCS module. The latter greatly impacts the transaction validation process at this level, which contributes to significant CAPEX and OPEX gains.

We implement BCOOL modules in the Network Simulator 3.26 (NS-3.26) [50] and practically evaluate its performance using real traffic data from the Italian city of Bologna [51]. For our experiments, we use the Simulation of Urban Mobility (SUMO) [52] as a simulation methodology for the Bologna dataset [51]. The Bologna dataset covers a typical day traffic between 8:00 am and 9:00 am (rush hour), with more than 22000 vehicles; in our experiments, we mainly focus on routes of congested regions located along the Bologna ringway. The implementation of BCOOL edge and cloud modules is based on the OFSwitch13 module [91] in NS-3.26. This module provides support for the OpenFlow protocol version 1.3 [23]. The communication between controllers and switches is realized over standard ns3 channels and devices. We conducted extensive simulation experiments to evaluate and compare the performance of HOMLRC based on Broadtrip to the performance of HOMLRC based on three other communication protocols, widely used in vehicular environments, namely Counter-Based Scheme (CBS) [48], Power-Aware Message Propagation Algorithm (PAMPA) [49] and Flooding (FLOOD). It worths mentioning that all the functionalities have been implemented based on IEEE 802.11 MAC, with TwoRay-Ground propagation model for traffic propagation in the considered highway scenario. The performance of BCOOL HOMLRC-BroadTrip module is evaluated in terms of the following metrics:

- **Efficiency** that is defined by the forward ratio, i.e., the number of nodes that forward a broadcast message m divided by the number of nodes in the network.

- **Reliability** that is defined by the delivery ratio, i.e., the number of nodes that deliver the message divided by the number of nodes in the network.
- **Average Throughput** that is defined by the average number of bytes received successfully by the receivers per time unit.
- **Average Delay** that is defined by the average time elapsed between the broadcasting of a message by a source and the delivery of this message by all vehicles in the network.
- **OffChain Contract Execution Frequency** that is defined by the number of times the Offchain-based MLR contract is invoked which is function of the network performance reliability at the OnChain contract level.

TABLE 1. Simulation parameters.

Parameters	Value
Type and length of road	Highway, 8 km
Number of lanes	3 lanes, same direction
Maximum speed	10-20 m/s
Number of vehicles per cluster	50, 75, 100, 125 and 150 vehicles
Transmission range	200 m - 250 m
Simulation time	500 s
Warm-up period	150 s
Simulation runs	10

A. SIMULATION PARAMETERS

Table 1 shows the simulation parameters commonly used in the literature [92]. The considered highway scenario is 8 km long with three curved lanes starting from ''Viale Giambattista Ercolani'' highway, passing by ''Viale Angelo Masini'', ''Viale Aldini'' and ''Viale Enrico Panzacchi'' highways, and ending at ''Viale Giovani Gozzadini'' highway. There are 8 RSUs beside the highways and they are 1 km apart from each other. Each RSU is connected to a switch node using CSMA links and each switch node is connected to a controller node using OpenFlow channels. Each pair of switch nodes connected to a pair of RSUs, which are located 1 km apart from each other, are controlled by a switch node that is connected to a controller node in an OpenFlow network. The transmission range of the vehicles is 200 m, whereas, the velocity and density of the vehicles are set based on different scenarios.

To simulate our module, all vehicles send packets of size 1000 bytes. We choose the maximum waiting time *mwt* to be 1 second. The average distance between vehicles varies between 8 and 10 meters. The simulation lasts for 500 s. The first 150 s are used to initialize the simulations and to make sure that the vehicles and the network have become stable. The simulation of each data point is repeated 10 times with different random seeds. We performed a statistical analysis by averaging the simulated values to a mean value that we compare with the benchmark strategies. We also compare the Confidence Interval (CI) and the Standard Deviation (STDEV) of the obtained simulation results for each performance metric.

B. SIMULATION RESULTS

Hereafter, we present the performance of HOMLRC-BroadTrip in terms of the aforementioned metrics, in critical data congestion environments, characterized by an increasing number of messages and nodes, as well as high message rates represented by the Inter-Packet Interval (IPI) parameter. IPI defines the interval that separates the sending of messages by each vehicle in the considered clusters. IPI values range from 10 μs and 100 ms to represent realistic critical data congestion scenarios. The lower is the IPI, the higher is the message rate. All the simulation results with a 95% confidence interval are represented in Figures 5, 6 and 7.

Figure 5 contrasts the variation of the delivery ratio, forward ratio, average throughput, and delivery delay while varying the density of clustered nodes along with the number of messages. The IPI is set to 100 ms.

Figure 6 shows the variation of the delivery ratio, forward ratio, average throughput, and delivery delay while varying the IPI and the transmission range of vehicles in a high node cluster density.

Figure 7 illustrates the variation of the OffChain contract execution frequency as function of the IPI and the number of messages for different communication protocols in low, medium, and high node cluster density environments.

The graphs of Figure 5 contrast the variation of four performance metrics namely the efficiency (forward ratio), reliability (delivery ratio), average throughput and delay as function of the node cluster densities where the distance between nodes varies between 8 and 10 meters and where each vehicle sends 10 and 30 messages, one message every 100 ms, in a range of 200 m.

As it can be seen from the simulation results, the forward ratio, delivery delay and throughput of our HOMLRC-BroadTrip strategy increases as long as the node cluster density increases. On the other hand, and in terms of reliability, Fig. 5(b) shows that HOMLRC-BroadTrip reaches 100% delivery rate for low and medium node cluster densities and 96% to 80% for high node cluster density. Nevertheless, the performance of other mechanisms namely, HOMLRC-CBS, HOMLRC-PAMPA and HOMLRC-FLOOD is far from reaching HOMLRC-BroadTrip's potential performance for all considered node densities and network sizes.

In low density node cluster (50-75 nodes) and when each vehicle individually sends 30 messages, one message every 100 ms, HOMLRC-BroadTrip successfully reaches 100% delivery rate with extremely higher throughput values, lower delay values and 0% retransmission (see Fig. 5(a)). In terms of forward ratio, HOMLRC-BroadTrip beats HOMLRC-CBS, HOMLRC-PAMPA and HOMLRC-FLOOD by 100% forward gain. In terms of delivery ratio, HOMLRC-BroadTrip beats HOMLRC-CBS, HOMLRC-PAMPA and HOMLRC-FLOOD by 53% to 64%, 57% to 65% and 56%

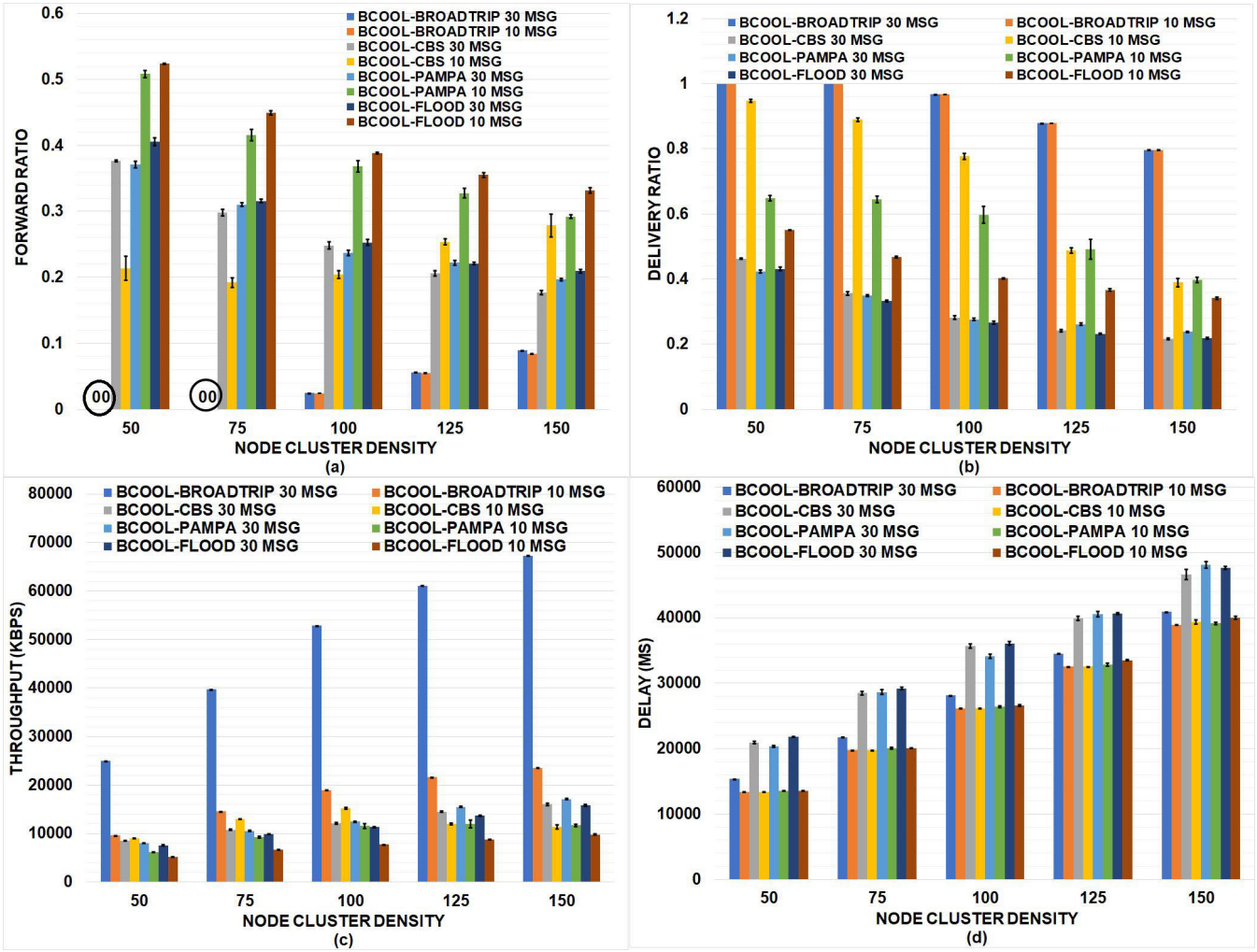

FIGURE 5. Simulation results of BCOOL-BroadTrip, with a 95% confidence interval, compared with four strategies - (a) Forward ratio, (b) Delivery ratio and (c) Throughput ratio, (d) Average delay. The IPI is 100 ms and R=200 m.

to 66% delivery gain respectively. In terms of throughput, we see that HOMLRC-BroadTrip provides high throughput gains over existing benchmark strategies. It outperforms them by 66% to 73%, 68% to 73.44% and 69% to 75% throughput gain respectively. In terms of delay, and as it is represented in Fig. 5(d), HOMLRC-BroadTrip successfully provides significant delay gains over other strategies. It surpasses them by 26% to 23%, 24.35% and 29% to 25.5% delay gains respectively.

In medium density node cluster (100 nodes), HOMLRC-BroadTrip average delivery ratio slightly decreases to reach 96.8% while maintaining a higher average throughput and lower delay values. In terms of forward ratio, HOMLRC-BroadTrip beats HOMLRC-CBS, HOMLRC-PAMPA and HOMLRC-FLOOD by 90%, 89% and 90% respectively. In terms of delivery ratio, HOMLRC-BroadTrip surpasses HOMLRC-CBS, HOMLRC-PAMPA and HOMLRC-FLOOD by an average gain of 70%, 71% and 72% respectively. In terms of average throughput values, HOMLRC-BroadTrip outperforms HOMLRC-CBS, HOMLRC-PAMPA and HOMLRC-FLOOD by 77%, 76% and 78% throughput gain respectively. HOMLRC-BroadTrip delivers a huge number of messages in a shorter time compared to other strategies which fail to deliver a comparable number of messages. It beats HOMLRC-CBS, HOMLRC-PAMPA and HOMLRC-FLOOD by 21%, 17.5% and 22% delay gains respectively.

In high density node cluster (125-150 nodes), the forward ratio of HOMLRC-BroadTrip slightly increases and beats HOMLRC-CBS, HOMLRC-PAMPA and HOMLRC-FLOOD forward ratios by 72.86%, 74.84% and 74.7% respectively. The average delivery ratio of HOMLRC-BroadTrip slightly decreases and reaches 80% delivery ratio. However, it continues to surpass the average delivery ratios of other comparable strategies. Globally, it beats HOMLRC-CBS, HOMLRC-PAMPA and HOMLRC-FLOOD by 72%, 70% and 73% respectively. In terms of throughput, we see in Fig. 5(c) that HOMLRC-BroaTrip average throughput

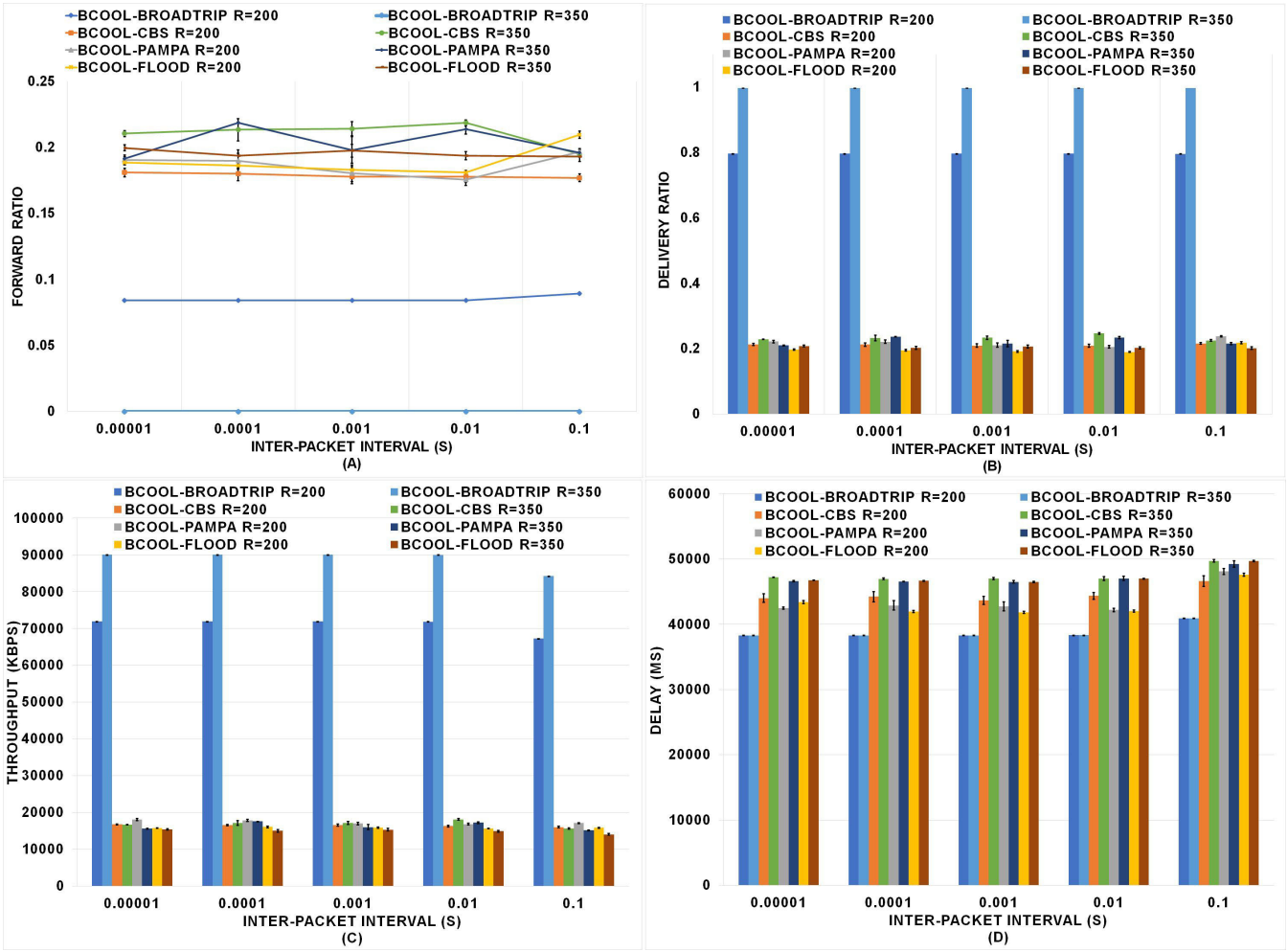

FIGURE 6. Simulation results of BCOOL-BroadTrip in high node cluster density (150 nodes), with a 95% confidence interval, compared with four strategies - (a) Forward ratio, (b) Delivery ratio and (c) Throughput ratio, (d) Average delay. The number of messages is 30 messages.

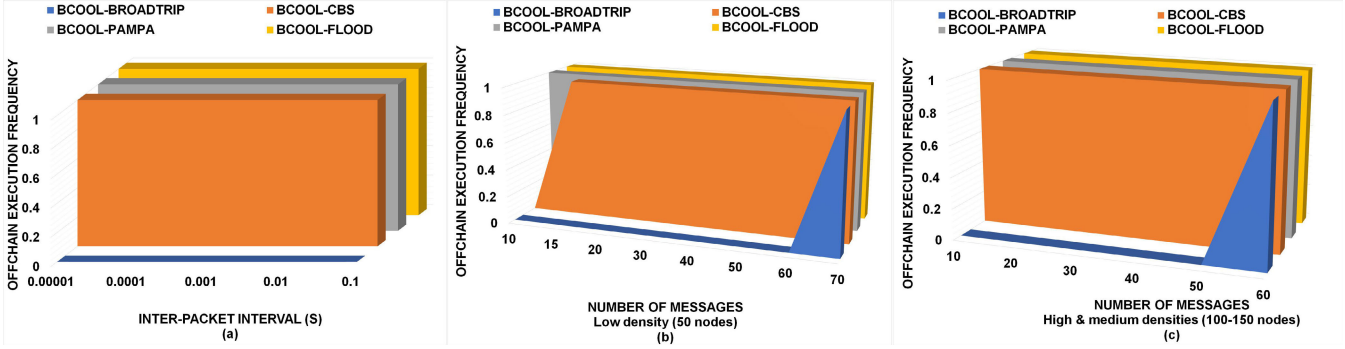

FIGURE 7. Simulation results of the Offchain Contract Execution Frequency function of the Inter-Packet Interval and the number of messages.

significantly increases and surpasses other strategies by 76%, 74% and 77% respectively. In terms of delivery delay, HOMLRC-BroadTrip values keep being lower than those of other strategies even in high density node cluster. HOMLRC-BroadTrip saves 13%, 14.9% and 15% delay of the other three strategies' average delivery delays.

On the other hand and when each vehicle sends 10 messages, one message every 100 ms, we clearly see that HOMLRC-BroadTrip performance still dominates other strategies' performance. However, we see that when we decrease the number of messages that each vehicle sends, the performance of the benchmark strategies increases for low

to medium densities and slightly decreases as the node cluster density increases.

In fact, in low density node cluster (50-75 nodes), HOMLRC-BroadTrip still beats other strategies by 100% in terms of forward ratio and successfully surpasses HOMLRC-CBS, HOMLRC-PAMPA and HOMLRC-FLOOD by 5% to 10%, 35% and 44% to 53% in terms of average delivery gain, respectively. The latter systematically reflects on the throughput and delay. In fact, HOMLRC-BroadTrip again outperforms other strategies' average throughput by 5% to 10%, 35% to 36% and 45% to 54%, respectively. Moreover, it takes 98% to 99% of the delivery delay of other strategies to deliver a huge amount of messages compared to other strategies that fail to reach the same performance.

In medium density node cluster (100 nodes) and in terms of forward ratio, HOMLRC-BroadTrip beats other strategies by 88% to 93% respectively. In terms of delivery ratio and throughput, HOMLRC-BroadTrip surpasses other strategies by 19.6%, 38% and 58.4% delivery and throughput gains. Moreover, HOMLRC-BroadTrip takes 99.99% to 98% of the delivery delay of other benchmark strategies to achieve good performance that outperforms other strategies' performance.

In high density node cluster (125-150 nodes), we see that HOMLRC-BroadTrip shows a forward gain of 78% to 69%, 83% to 71% and 84% to 74% over other benchmark strategies' forward ratios. Moreover, it outperforms the delivery performance of other strategies by 45% to 51%, 44% to 50% and 60% to 58% gain respectively and it achieves those performance in 97% to 99% of the delivery delay of other strategies.

The reason behind BCOOL-BroadTrip's high performance in this critical vehicular environment, where each vehicle continuously and periodically sends a message each 100 ms, lies in the fact that BCOOL-BroadTrip relies on a smart and powerful combination of Non Forward Zone (NFZ), network coding [53] and location-based wait and count mechanisms [49]. Network coding mixes two messages from opposite directions and sends them at the price of one message which lowers the number of messages forwarded throughout the network and strengthens the messages delivery thanks to the location-based wait-and-count mechanism that selects forwarders based on their distances from senders. Vehicles located inside the NFZ decode the received messages but do not establish a retransmission schedule to forward those messages, they only decode, deliver and stay silent. The combination of those mechanisms contributes to a successful and reliable dissemination of messages for any node density even in critical conditions as shown through simulations.

Other strategies fail to achieve a similar performance in the two considered critical vehicular scenarios for all node cluster densities.

In fact, the performance of HOMLRC-CBS, HOMLRC-PAMPA and HOMLR-FLOOD increased when we decreased the amount of messages to 10 messages per vehicle. However, this performance dramatically decreased when we increased the number of messages to 30 message per vehicle. In addition, even if we decreased the number of messages, the performance of benchmark strategies still decreases as the network size increases. This led to a significant drop of messages, huge delivery delays, low throughput values and huge overhead over the network.

Globally, we notice that increasing the number of messages along with the node cluster densities yield instability in the considered challenging vehicular scenarios. This instability arises consistently when the increasing number of vehicles back off for a random amount of time and simultaneously contend for the shared wireless channel in order to send/forward their instant messages every 100 ms. The large number of users that instantly and simultaneously compete to send their messages cause massive message collisions after reaching the back off limit which dramatically increases message losses and delivery delays. This network instability proves that those benchmark protocols namely CBS, PAMPA and FLOOD are not suitable solutions for critical vehicular scenarios because they are easily prone to data congestion.

This network instability is clearly apparent in the statistical analysis of the delay metric that we display in Table 2. In terms of this statistical analysis, we see that when we increase the number of messages, the reliability of HOMLRC-BroadTrip maintains its optimality compared to the reliability of other benchmark strategies that drops drastically (see Figure 5(b)). This fact greatly impacts the precision of the CI of HOMLRC-BroaTrip which is stable and optimal compared to the CI's precision of other benchmark strategies. More specifically, the low delivery ratio of other strategies and their increasing number of dropped messages introduce randomness and increase heterogeneity in the considered critical vehicular scenarios which systematically widen the correspondent CIs and STDEVs of benchmark strategies. This confirms the outstanding performance of our HOMLRC-BroadTrip strategy.

To further demonstrate the potential performance of our HOMLRC-BroadTrip strategy in critical data congestion environments, we evaluated the impact of varying the transmission range and increasing the message rate on the considered metrics. Figure 6 contrasts the variation of the forward ratio, delivery ratio, throughput and delivery delay as function of extremely low IPIs which correspond to high message rates when the transmission range is equal to 200 m and 350 m, respectively, in high density node cluster (150 nodes).

As it can be seen from the simulation results, the performance of HOMLRC-BroadTrip is far above the performance of other benchmark strategies. Contrary to HOMLRC-BroadTrip, all state-of-the-art strategies fail to deliver the same number of messages in the considered critical data congestion scenarios when the IPI values range from 10 μs to 100 ms. Although we increased the value of the transmission range, the benchmark strategies' behaviour remains stagnant. However, the performance of HOMLRC-BroadTrip increased significantly when the vehicles' transmission range increased.

TABLE 2. Statistical analysis: delay (ms).

Node cluster Density	50	75	100	125	150
Delay: BCOOL-BROAD (30 msgs)	15364.6 ms	21752.5 ms	28140.4 ms	34528.5 ms	40916.2 ms
Delivery Ratio	**1**	**1**	**0.9679128**	**0.8784785**	**0.7963149**
CI(delay)	**0.727507842**	**1.140976648**	**1.466702744**	**1.876522649**	**2.259396367**
STDEV(delay)	**1.173787791**	**1.840893503**	**2.366431913**	**3.027650354**	**3.645392831**
Delay: BCOOL-BROAD (10 msgs)	13364.6 ms	19752.5 ms	26140.4 ms	32528.5 ms	38916.2 ms
Delivery Ratio	**1**	**1**	**0.967828**	**0.87845**	**0.7963736**
CI(delay)	**0.727507842**	**1.140976648**	**1.466702744**	**1.876522649**	**2.259396367**
STDEV(delay)	**1.173787791**	**1.840893503**	**2.366431913**	**3.027650354**	**3.645392831**
Delay: BCOOL-CBS (30 msgs)	20922.3 ms	28485.4 ms	35712.1 ms	39913.2 ms	46610.5 ms
Delivery Ratio	**0.4629438**	**0.3563482**	**0.281828**	**0.2416417**	**0.2160258**
CI(delay)	**162.667578**	**299.2096422**	**310.771359**	**301.2934656**	**807.868437**
STDEV(delay)	**262.4538266**	**482.7557929**	**501.40989**	**486.1179098**	**1303.444518**
Delay: BCOOL-CBS (10 msgs)	13368.7 ms	19753.3 ms	26141 ms	32528.5 ms	39369.5 ms
Delivery Ratio	**0.94765197**	**0.8902028**	**0.7771735**	**0.487849**	**0.3898825**
CI(delay)	**5.755534719**	**1.825796429**	**1.82462717**	**1.876522649**	**304.7269198**
STDEV(delay)	**9.286190464**	**2.945806813**	**2.943920289**	**3.027650354**	**491.6575705**
Delay: BCOOL-PAMPA (30 msgs)	20311.6 ms	28650.7 ms	34117.1 ms	40560 ms	48080.6 ms
Delivery Ratio	**0.4233187**	**0.3497251**	**0.2761421**	**0.2615687**	**0.2379475**
CI(delay)	**136.3950314**	**390.7771552**	**320.8073097**	**454.2928002**	**479.4371046**
STDEV(delay)	**220.0647379**	**630.4941713**	**517.6022604**	**732.9726386**	**773.5413801**
Delay: BCOOL-PAMPA (10 msgs)	13531.7 ms	20046.9 ms	26418.9 ms	32856 ms	39144 ms
Delivery Ratio	**0.6483375**	**0.6450529**	**0.5974687**	**0.492314**	**0.3971806**
CI(delay)	**88.94966884**	**146.940507**	**153.9343314**	**215.589109**	**156.1107178**
STDEV(delay)	**143.5146528**	**237.0791945**	**248.3632868**	**347.8393627**	**251.8747484**
Delay: BCOOL-FLOOD (30 msgs)	21795.3 ms	29190.2 ms	36090.2 ms	40643.1 ms	47604.4 ms
Delivery Ratio	**0.4310313**	**0.3331185**	**0.2659549**	**0.2315117**	**0.218179**
CI(delay)	**77.99794818**	**180.8038418**	**279.2448212**	**121.6562379**	**240.8538346**
STDEV(delay)	**125.8447456**	**291.7155388**	**450.5438195**	**196.2846289**	**388.6023961**
Delay: BCOOL-FLOOD (10 msgs)	13556.9 ms	20084.9 ms	26572.5 ms	33513.3 ms	40002.6 ms
Delivery Ratio	**0.5513376**	**0.4674501**	**0.4020609**	**0.366396**	**0.3411152**
CI(delay)	**14.97930605**	**36.18415325**	**117.865487**	**106.7216138**	**196.8499986**
STDEV(delay)	**24.16816087**	**58.3808378**	**190.1684925**	**172.1885594**	**317.6049958**

In fact, when we set the vehicles' transmission range to 200 m and vary the IPI from 10 μs to 100 ms, we see that in terms of forward ratio, HOMLRC-BroadTrip outperforms other benchmark strategies, namely HOMLRC-CBS, HOMLRC-PAMPA and HOMLRC-FLOOD by 49% to 53%, 51% to 55% and by 53% to 57% respectively. In terms of delivery ratio, HOMLRC-BroadTrip demonstrates higher reliability compared to other strategies. It surpasses HOMLRC-CBS by 72% to 73%, HOMLRC-PAMPA by 70% to 74% and HOMLRC-FLOOD by 72% to 76%. Moreover, HOMLRC-BroadTrip achieves extremely higher throughput values compared to other strategies. It outperforms HOMLRC-CBS, HOMLRC-PAMPA and HOMLRC-FLOOD by 76% to 77%, 74.88% to 76.55% and 76% to 78.27% respectively. Additionally, HOMLRC-BroadTrip delivery delay represents 86% to 87% of the average delay of HOMLRC-CBS, 85% to 90% of the average delay of HOMLRC-PAMPA and 85% to 91% of the average delay of HOMLRC-FLOOD.

On the other hand, when we increase vehicles' transmission range to 350 m and keep varying the IPI from 10 μs to 100 ms, we see that HOMLRC-BroadTrip provides a forward gain of 100% over other benchmark strategies for all values of IPI. In terms of delivery ratio, HOMLRC-BroadTrip succeeds to deliver all messages with high average throughput values and low delivery delays. It surpasses HOMLRC-CBS, HOMLRC-PAMPA and HOMLRC-FLOOD by a delivery gain of 75% to 77%, 76% to 78% and 80% respectively. It provides a throughput gain of 81%, 82% and 83% over the average throughput of HOMLRC-CBS, HOMLRC-PAMPA and HOMLRC-FLOOD respectively. Moreover, HOMLRC-BroadTrip achieves 100% delivery in a short delay compared to other strategies. It saves 17.6% to 19%, 16.8% to 18.5% and 18% of the delay of HOMLRC-CBS, HOMLRC-PAMPA and HOMLRC-FLOOD respectively.

As we can see, increasing the transmission range represents an excellent alternative for HOMLRC-BroadTrip to increase its reliability and overall performance even in critical

data congestion environments. Nevertheless, this alternative does not impact the performance of other strategies which remains stagnant in regards of the change in the transmission range. It is clear that the performance of those strategies, in those critical network settings, is relentlessly blocked. This persistent behaviour is obviously due to massive collisions that occur in shared wireless channels when all vehicles concurrently contend to instantly send out their messages over the physical medium. This incapacity to handle critical data congestion yields to drastic performance losses which directly and negatively impact the CAPEX and OPEX cost of BCOOL HOMLRCS and KRF-ODSFCS modules. More specifically, the inability of the benchmark strategies to tackle critical data congestion represents one of the chief obstacles that disrupt the normal functioning of BCOOL. This obstacle engenders huge time and resource consumption at different levels of the architecture. For instance, the low reliability of benchmark strategies significantly and badly increases the MLR Offchain's workload which ossifies the mining process, incurs long provision delays at different architecture layers and intensifies the system's vulnerability to security attacks.

Figure 7(a) represents the variation of the OffChain Contract execution frequency as function of the IPI when the number of messages that each vehicle sends out in high node cluster density is 30 messages in 350 m transmission range. Figures 7(b) and 7(c) illustrate the variation of the OffChain Contract execution frequency as function of the number of messages when the IPI is set to 100 ms in low, medium and high node densities.

It is worth noting that the OffChain contract is invoked when the network performance degrades at the Onchain contract level which corresponds to a low delivery rate.

Accordingly, and as shown in figure 7(a), HOMLRC-BroadTrip maintains a 100% message delivery rate even when each vehicle sends out 30 messages with extremely high message rates which range from 10 μs to 100 ms. We clearly see that HOMLRC-BroadTrip performance does not invoke the process of the OffChain contract contrary to other benchmark strategies' relentless bad performance which continuously trigger the Offchain congestion prediction contract to predict and avoid the occurrence of congestion.

On the other hand, when we set the IPI to 100 ms and vary the number of messages in all node cluster densities, we see that increasing the number of vehicle messages impacts the running of the Offchain execution frequency as we proved it in the computational complexity analysis section. In fact, in low node cluster density (50 nodes), HOMLRC-BroadTrip maintains a high reliability rate until the number of messages, that each vehicle sends out, reaches 70 messages. This is in contrast with medium and high node cluster densities (100-150 nodes) whose delivery performance started decreasing when the number of messages reaches 60 messages. On the other hand, HOMLRC-CBS delivery performance started dropping, in low node cluster density (50 nodes), when vehicle messages reaches only 15 messages. However, other strategies exhibit very low delivery performance for all increasing number of messages in all node cluster densities in the considered critical network congestion setting.

This is due to excessive collisions that occur when the large number of nodes simultaneously contend over the shared physical medium to transmit their instantaneous messages. Those strategies' low performance systematically and recurrently triggers the Offchain contract strategy to predict and avoid congestion when their delivery performance and message loss is less than 75% and greater than 25%, respectively [20]. This makes BCOOL consume huge computing resources which creates bottlenecks at different levels of BCOOL architecture and dramatically increases CAPEX and OPEX expenditures.

It first and automatically increases the workload of the OffChain congestion prediction contract which runs indefinitely to predict and avoid the happening of data congestion. This workload systematically biases the prediction results' accuracy and badly delays the reception of the prediction results (if correctly issued) by clustered vehicles. Second, this makes miners wait for the Offchain contract results in order to reach consensus and add transactions to the chain of blocks. Third, the consequences of those strategies' poor performance badly damage the operating of the upper KRF-ODSFC strategy. In fact, the extremely low reliability of those strategies precludes the full collection of user's requests at the Offchain K-means clustering contract strategy. This results in inconsistent VNF prediction decisions that do not match users' QoS priority requests. This in addition to inducing long provision delays at both SFC prediction process and the SFC transactions mining process.

Unlike those strategies' performance, BCOOL-BroadTrip exhibits a strong resilience against data congestion which efficiently smooths the functioning of its KRF-ODSFC strategy. In fact, in transient and time-varying traffic congestion, BCOOL-BroadTrip flexibly adapts to traffic changes. The HOMLRC-BroadTrip Offchain contract is completely switched off for extremely low values of IPI which corresponds to high message rates in high cluster node densities when the number of messages is below 50 messages. This Offchain contract is only invoked when the number of messages reaches 60 and 70 messages in low medium and high cluster densities, respectively. This presents a significant gain over other strategies, alleviates BCOOL modules security concerns, and speeds up the consensus process. Moreover, and in addition to this gain outcomes, the high reliability of BCOOL-BroadTrip consolidates the full collection of user requests at the Offchain K-means contract which contributes to reliable and effective VNF placement decisions. Using the BroadTrip protocol, the provision delay of those reliable decisions (see figure 8) only increases linearly with user requests which represents another significant CAPEX and OPEX gains compared to traditional VNF placement strategies [34] that rely on high complexity BACON heuristic solutions.

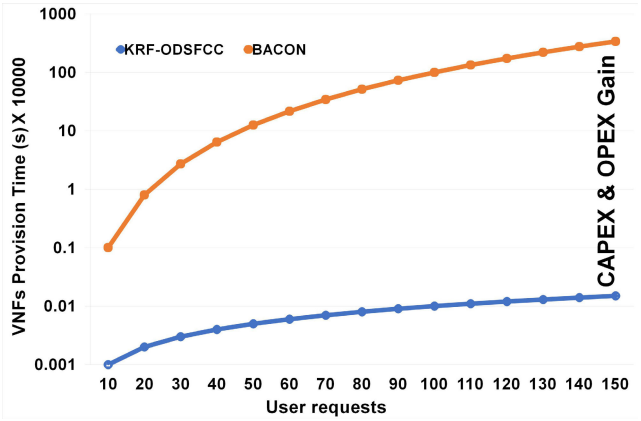

FIGURE 8. CAPEX and OPEX Gain.

C. ASSESSMENT

As predicted, Broadtrip is the most convenient protocol for our proposed BCOOL's cloud/edge-based On/Off-chain smart contract modules. Contrary to other congestion control/prediction strategies that follow the same and classic trend of controlling/predicting congestion while relying on unknown communication protocols [2], [20], BCOOL's computational complexity analysis and simulations proved that communication protocols rule the functioning of data congestion prediction/control systems and represent a fundamental input to efficiently and accurately control the congestion. Fully ignoring the importance of the efficiency and reliability of the deployed data dissemination protocols, while predicting data congestion at different network levels, only ossifies the network infrastructure which biases the congestion prediction results, amplifies CAPEX and OPEX costs and creates bottlenecks at different data congestion prediction network entities.

Extensive simulations proved that HOMLRC-BroadTrip's strongly handles data congestion in extremely challenging data congestion environments which provides significant CAPEX and OPEX gain. It saves BCOOL's computing resources and optimizes its whole mining process at different infrastructure levels, which validates the effectiveness of our system's implementation plan. In fact, using this protocol, SDN controllers mainly monitor the congestion using Onchain contracts that only verify the rate at which messages are exchanged throughout the network and deliver smooth traffic flow to upper layers. The frequency at which the Offchain contract is to be triggered is strictly minimized thanks to the high and flexible performance of BroadTrip in challenging data congestion environments. This workload reduction at the Offchain level improves the quality of the accuracy of prediction results and speeds up the provision of VNF prediction decisions. As demonstrated, the performance of other strategies severely struggles in challenging data congestion environments characterized by an increasing number of nodes, messages and high message rates. The latter ossifies the operating of BCOOL cloud-based On/Offchain modules, makes them prone to data vulnerabilities which renders the data congestion prediction results, if correctly issued, meaningless and the process useless.

An interesting finding is that the use of BroadTrip contributes to the provision of fast and accurate congestion prediction decisions that reach vehicles ahead of time. Another interesting finding is that this reduction in the execution frequency of BCOOL-HOMLR Offchain contract, in critical data congestion environments, fully contributes to the acquisition of smooth, real-time unlabeled traffic by the K-means Offchain contract at the OSS/BSS level which fastens and refines the prediction of SFC deployment decisions. This also systematically contributes to a significant resource saving in the transaction validation process and obviously to a considerable CAPEX and OPEX gain.

VII. CONCLUSION

In this paper, we focused on a very important but seldom studied problem that is the lack of trustworthy Blockchain congestion prediction systems. We proposed a novel, flexible, dynamic, consistent, resilient and unique Blockchain Congestion ContrOL (BCOOL) system for vehicular networks. BCOOL relies on three novel modules. The first module is BCOOL's building block that aims at dynamically and reliably managing recent message and vehicle trustworthiness, at the edge of the network, and at preventing distrustful vehicles from acquiring access to the infrastructure. The second novel module immutably, and dynamically learns in real-time about congestion while relying on an efficient and reliable broadcasting protocol to accurately and fastly predict the occurrence of congestion, ahead of time, and to consequently lighten the transaction validation process. The third novel KRF-ODSFCS module proactively and accurately predicts VNF placements and their chaining order in the context of SFCs while considering both dynamic user QoS priority requests, received using the HOMLRC strategy, and the requirements of NFV applications. BCOOL dynamically and immutably records message, vehicle trustworthiness and prediction transactions in the distributed ledger at the edge of the network with a linear time complexity.

The Byzantine resilience and attack mitigation strategies proved that BCOOL not only satisfied the security requirements of vehicular networks but also provided countermeasures to withstand failures, DDOS, double spending and the majority (51%) consensus attacks.

Taking into account a real-world mobility scenario, the mining performance of BCOOL based on the BroadTrip protocol was evaluated and compared to other strategies in critical data congestion environments. The comparison showed that our proposed system outperformed other strategies in the considered challenging mobility scenarios with significant CAPEX/OPEX gains. Overall, the failure model, threat defense strategies and complexity analysis along with the obtained results proved that BCOOL-BroadTrip efficiently copes with simultaneous QoS priority user requests and provides fast, accurate and consistent predictive decisions, ahead of time, while alleviating failures and security

concerns and lowering the whole infrastructure-based block generation delay.

As future research directions, we plan to relax some of the constraints introduced in the design of the proposed system and investigate their impact on the performance of BCOOL using real-world evaluations.

REFERENCES

[1] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.

[2] V. Hassija, V. Gupta, S. Garg, and V. Chamola, "Traffic jam probability estimation based on blockchain and deep neural networks," *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 3, 2020, doi: 10.1109/TITS.2020.2988040.

[3] T. Taleb, A. Ksentini, M. Chen, and R. Jantti, "Coping with emerging mobile social media applications through dynamic service function chaining," *IEEE Trans. Wireless Commun.*, vol. 15, no. 4, pp. 2859–2871, Apr. 2016.

[4] R. Shrestha, R. Bajracharya, A. P. Shrestha, and S. Y. Nam, "A new type of blockchain for secure message exchange in VANET," *Digit. Commun. Netw.*, vol. 6, no. 2, pp. 177–186, May 2020, doi: 10.1016/j.dcan.2019.04.003.

[5] P. K. Sharma, M.-Y. Chen, and J. H. Park, "A software defined fog node based distributed blockchain cloud architecture for IoT," *IEEE Access*, vol. 6, pp. 115–124, 2018, doi: 10.1109/ACCESS.2017.2757955.

[6] S. Maaroufi and S. Pierre, "OnlineCruise: An online social grouping strategy for vehicular social networks," in *Proc. 5th ACM Symp. Develop. Anal. Intell. Veh. Netw. Appl.*, Nov. 2015, pp. 67–70.

[7] Y. Lu, Q. Tang, and G. Wang, "On enabling machine learning tasks atop public blockchains: A crowdsourcing approach," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2018, pp. 81–88.

[8] M. S. Bonfim, K. L. Dias, and S. F. L. Fernandes, "Integrated NFV/SDN architectures: A systematic literature review," *ACM Comput. Surv.*, vol. 51, no. 6, pp. 114:1–114:39, 2019, doi: 10.1145/3172866.

[9] J. Yang, S. He, Y. Xu, L. Chen, and J. Ren, "A trusted routing scheme using blockchain and reinforcement learning for wireless sensor networks," *Sensors*, vol. 19, no. 4, p. 970, Feb. 2019.

[10] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F.-Y. Wang, "Blockchain-enabled smart contracts: Architecture, applications, and future trends," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 11, pp. 2266–2277, Nov. 2019.

[11] H. Hasrouny, A. E. Samhat, C. Bassil, and A. Laouiti, "VANet security challenges and solutions: A survey," *Veh. Commun.*, vol. 7, pp. 7–20, Jan. 2017, doi: 10.1016/j.vehcom.2017.01.002.

[12] P. K. Sharma, S. Singh, Y.-S. Jeong, and J. H. Park, "DistBlockNet: A distributed blockchains-based secure SDN architecture for IoT networks," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 78–85, Sep. 2017.

[13] M. B. Younes and A. Boukerche, "SCOOL: A secure traffic congestion control protocol for VANETs," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2015, pp. 1960–1965.

[14] M. B. Younes, "Secure traffic efficiency control protocol for downtown vehicular networks," *IJ. Netw. Secur.*, vol. 21, no. 3, pp. 511–521, 2019.

[15] U. Lee, R. Cheung, and M. Gerla, *Vehicular Net-works: From Theory to Practice*. London, U.K.: CRC Press, 2008, ch. 1.

[16] A. Holzer, S. Maaroufi, and S. Pierre, "BROADTRIP: Broadcast for transit in platoons," in *Proc. IEEE 7th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2011, pp. 301–306.

[17] S. Maaroufi, "A dynamic messaging architecture for vehicular social networks," Ph.D. dissertation, Ecole Polytechnique de Montreal, Montreal, QC, Canada, 2016.

[18] A. Holzer, S. Maaroufi, and S. Pierre, "DYMES: A dynamic messaging service for VANETs," in *Proc. IEEE 6th Int. Conf. Wireless Mobile Comput., Netw. Commun.*, Oct. 2010, pp. 513–520.

[19] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Stat. Probab.*, 1967, pp. 281–297.

[20] N. Taherkhani, "Congestion control in vehicular adhoc networks," Ph.D. dissertation, Ecole Polytechnique de Montreal, Montreal, QC, Canada, 2015.

[21] N. John, C. J. Nachtsheim, J. Neter, and W. Li, *Applied Linear Statistical Models*, vol. 4. Chicago, IL, USA: Irwin, 1996.

[22] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[23] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[24] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proc. 5th Annu. ACM/IEEE Int. Conf. Mobile Comput. Netw. (MobiCom)*, 1999, pp. 151–162.

[25] D. Wu, D. Zhang, and L. Sun, "An aggregate parameter for congestion detection in VANETs," in *Proc. 5th Int. Conf. Inf. Comput. Sci.*, Liverpool, U.K., Jul. 2012, pp. 95–98.

[26] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an optimized blockchain for IoT," in *Proc. IEEE/ACM 2nd Int. Conf. Internet-Things Design Implement. (IoTDI)*, Apr. 2017, pp. 173–178.

[27] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, "BlockChain: A distributed solution to automotive security and privacy," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 119–125, Dec. 2017.

[28] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2017, pp. 618–623.

[29] C. Qiu, F. R. Yu, F. Xu, H. Yao, and C. Zhao, "Blockchain-based distributed software-defined vehicular networks via deep Q-learning," in *Proc. 8th ACM Symp. Design Anal. Intell. Veh. Netw. Appl. (DIVANet)*, Montreal, QC, Canada, 2018, pp. 8–14.

[30] D. Zhang, F. R. Yu, and R. Yang, "Blockchain-based distributed software-defined vehicular networks: A dueling deep *Q*-learning approach," *IEEE Trans. Cognit. Commun. Netw.*, vol. 5, no. 4, pp. 1086–1100, Dec. 2019.

[31] Y.-T. Yang, L.-D. Chou, C.-W. Tseng, F.-H. Tseng, and C.-C. Liu, "Blockchain-based traffic event validation and trust verification for VANETs," *IEEE Access*, vol. 7, pp. 30868–30877, 2019, doi: 10.1109/ACCESS.2019.2903202.

[32] J. Li, G. Liang, and T. Liu, "A novel multi-link integrated factor algorithm considering node trust degree for blockchain-based communication," *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 8, pp. 3766–3788, 2017.

[33] S. Zemouri, S. Djahel, and J. Murphy, "A short-term vehicular density prediction scheme for enhanced beaconing control," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–7.

[34] H. Hawilo, M. Jammal, and A. Shami, "Network function virtualization-aware orchestrator for service function chaining placement in the cloud," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 643–655, Mar. 2019, doi: 10.1109/JSAC.2019.2895226.

[35] M. R. Spada, J. Perez-Romero, A. Sanchoyerto, R. Solozabal, M. A. Kourtis, and V. Riccobene, "Management of mission critical public safety applications: The 5G ESSENCE project," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Valencia, Spain, Jun. 2019, pp. 155–160.

[36] S. Lange, H.-G. Kim, S.-Y. Jeong, H. Choi, J.-H. Yoo, and J. W.-K. Hong, "Predicting VNF deployment decisions under dynamically changing network conditions," in *Proc. 15th Int. Conf. Netw. Service Manage. (CNSM)*, Halifax, NS, Canada, Oct. 2019, pp. 1–9.

[37] S. Lange, H.-G. Kim, S.-Y. Jeong, H. Choi, J.-H. Yoo, and J. W.-K. Hong, "Machine learning-based prediction of VNF deployment decisions in dynamic networks," in *Proc. 20th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Matsue, Japan, Sep. 2019, pp. 1–6.

[38] Y. Li and M. Chen, "Software-defined network function virtualization: A survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.

[39] A. Patel, M. Vutukuru, and D. Krishnaswamy, "Mobility-aware VNF placement in the LTE EPC," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2017, pp. 1–7.

[40] R. Mijumbi, J. Serrat, J.-L. Gorricho, S. Latre, M. Charalambides, and D. Lopez, "Management and orchestration challenges in network functions virtualization," *IEEE Commun. Mag.*, vol. 54, no. 1, pp. 98–105, Jan. 2016.

[41] H. Zhao, H. Yu, D. Li, T. Mao, and H. Zhu, "Vehicle accident risk prediction based on AdaBoost-SO in VANETs," *IEEE Access*, vol. 7, pp. 14549–14557, 2019.

[42] S. Rahman, T. Ahmed, M. Huynh, M. Tornatore, and B. Mukherjee, "Auto-scaling VNFs using machine learning to improve QoS and reduce cost," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[43] H.-G. Kim, D.-Y. Lee, S.-Y. Jeong, H. Choi, J.-H. Yoo, and J. W.-K. Hong, "Machine learning-based method for prediction of virtual network function resource demands," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Jun. 2019, pp. 405–413.

[44] N. Kulkarni, D. Mantri, P. Pawar, and N. R. Prasad, "Averaging based predictive modelling for traffic congestion in IoT," in *Proc. IEEE Global Conf. Wireless Comput. Netw. (GCWCN)*, Nov. 2018, pp. 49–53.

[45] S. Zemouri, S. Djahel, and J. Murphy, "An altruistic prediction-based congestion control for strict beaconing requirements in urban VANETs," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 12, pp. 2582–2597, Dec. 2019.

[46] R. Sathya and A. Abraham, "Comparison of supervised and unsupervised learning algorithms for pattern classification," *Int. J. Adv. Res. Artif. Intell.*, vol. 2, no. 2, pp. 34–38, 2013.

[47] H. Ye, L. Liang, G. Ye Li, J. 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.

[48] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," *IEEE/ACM Trans. Netw.*, vol. 14, no. 3, pp. 479–491, Jun. 2006.

[49] B. Garbinato, A. Holzer, and F. Vessaz, "Context-aware broadcasting approaches in mobile ad hoc networks," *Comput. Netw.*, vol. 54, no. 7, pp. 1210–1228, May 2010.

[50] G. F. Riley and T. R. Henderson, "The Ns-3 network simulator," in *Modeling and Tools for Network Simulation*. New York, NY, USA: Springer. 2010.

[51] L. Bedogni, M. Gramaglia, A. Vesco, M. Fiore, J. Harri, and F. Ferrero, "The Bologna ringway dataset: Improving road network conversion in SUMO and validating urban mobility via navigation services," *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5464–5476, Dec. 2015.

[52] M. Behrisch, L. B. Walz, J. Erdmann, and D. Krajzewicz, "SUMO–simulation of urban mobility: An overview," in *Proc. SIMUL*, 2011, p. 5560.

[53] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[54] W. Su, S.-J. Lee, and M. Gerla, "Mobility prediction and routing in ad hoc wireless networks," *Int. J. Netw. Manage.*, vol. 11, no. 1, pp. 3–30, 2001.

[55] W. Li and H. Song, "ART: An attack-resistant trust management scheme for securing vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 960–969, Apr. 2016.

[56] R. G. Machado and K. Venkatasubramanian, "Short paper: Establishing trust in a vehicular network," in *Proc. IEEE Veh. Netw. Conf.*, Dec. 2013, pp. 194–197.

[57] X. Shen, X. Cheng, R. Zhang, B. Jiao, and Y. Yang, "Distributed congestion control approaches for the IEEE 802.11p vehicular networks," *IEEE Intell. Transp. Syst. Mag.*, vol. 5, no. 4, pp. 50–61, Winter 2013.

[58] M. Torrent-Moreno, P. Santi, and H. Hartenstein, "Distributed fair transmit power adjustment for vehicular ad hoc networks," in *Proc. 3rd Annu. IEEE Commun. Soc. Sensor Ad Hoc Commun. Netw.*, vol. 2, Sep. 2006, pp. 479–488.

[59] P. K. Sahu, A. Hafid, and S. Cherkaoui, "Congestion control in vehicular networks using network coding," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 2736–2741.

[60] O. Chakroun and S. Cherkaoui, "Overhead-free congestion control and data dissemination for 802.11p VANETs," *Veh. Commun.*, vol. 1, no. 3, pp. 123–133, 2014.

[61] C.-W. Hsu, C.-H. Hsu, and H.-R. Tseng, "MAC channel congestion control mechanism in IEEE 802.11p/WAVE vehicle networks," in *Proc. IEEE Veh. Technol. Conf. (VTC Fall)*, Sep. 2011, pp. 1–5.

[62] H.-C. Jang and W.-C. Feng, "Network status detection-based dynamic adaptation of contention window in IEEE 802.11p," in *Proc. IEEE 71st Veh. Technol. Conf.*, May 2010, pp. 1–5.

[63] C. Sommer, S. Joerer, M. Segata, O. K. Tonguz, R. L. Cigno, and F. Dressler, "How shadowing hurts vehicular communications and how dynamic beaconing can help," *IEEE Trans. Mobile Comput.*, vol. 14, no. 7, pp. 1411–1421, Jul. 2015.

[64] S. Bai, J. Oh, and J.-I. Jung, "Context awareness beacon scheduling scheme for congestion control in vehicle to vehicle safety communication," *Ad Hoc Netw.*, vol. 11, no. 7, pp. 2049–2058, Sep. 2013.

[65] N. S. Nafi, R. H. Khan, J. Y. Khan, and M. Gregory, "A predictive road traffic management system based on vehicular ad-hoc network," in *Proc. Australas. Telecommun. Netw. Appl. Conf. (ATNAC)*, Nov. 2014, pp. 135–140, doi: 10.1109/ATNAC.2014.7020887.

[66] J. Pei, P. Hong, and D. Li, "Virtual network function selection and chaining based on deep learning in SDN and NFV-enabled networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Kansas City, MO, USA, May 2018, pp. 1–6.

[67] I. D. Alvarenga, G. A. F. Rebello, and O. C. M. B. Duarte, "Securing configuration management and migration of virtual network functions using blockchain," in *Proc. NOMS-IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2018, pp. 1–9.

[68] M. F. Franco, E. J. Scheid, L. Z. Granville, and B. Stiller, "BRAIN: Blockchain-based reverse auction for infrastructure supply in virtual network functions-as-a-service," in *Proc. IFIP Netw. Conf. (IFIP Netw.)*, May 2019, pp. 1–9.

[69] A. Wismadi, J. Soemardjito, and H. Sutomo, "Transport situation in Jakarta," I. Kutani, Ed., Japan, Tech. Rep. ERIA Res. Proj. Report 2012-29, 2013, pp. 29–58.

[70] *Road Traffic Injuries*, World Health Org., Geneva, Switzerland. Accessed: Feb. 7, 2020.

[71] A. Laghrissi and T. Taleb, "A survey on the placement of virtual resources and virtual network functions," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1409–1434, 2nd Quart., 2019.

[72] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 3rd Quart., 2017.

[73] P. Tyagi and D. Dembla, "Investigating the security threats in vehicular ad hoc networks (VANETs): Towards security engineering for safer on-road transportation," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2014, pp. 2084–2090.

[74] S. Deelers and S. Auwatanamongkol, "Enhancing K-means algorithm with initial cluster centers derived from data partitioning along the data axis with the highest variance," *Int. J. Comput. Sci.*, vol. 2, no. 4, pp. 247–252, 2007.

[75] R. A. Haraty, M. Dimishkieh, and M. Masud, "An enhanced k-Means clustering algorithm for pattern discovery in healthcare data," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 6, Jun. 2015, Art. no. 615740.

[76] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.

[77] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-means clustering algorithm," *Appl. Statist.*, vol. 28, no. 1, pp. 100–108, 1979.

[78] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[79] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, USA: Springer, 2009.

[80] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *J. Gen. Philos. Sci.*, vol. 39, no. 1, p. 5367, 2008.

[81] R. C. Merkle, "A digital signature based on a conventional encryption function," in *Proc. CRYPTO*, 1987, p. 1620.

[82] X. Wang, X. Zha, W. Ni, R. P. Liu, Y. J. Guo, X. Niu, and K. Zheng, "Survey on blockchain for Internet of Things," *Comput. Commun.*, vol. 136, pp. 10–29, Feb. 2019.

[83] R. Zhang, R. Xue, and L. Liu, "Security and privacy on blockchain," *ACM Comput. Surv.*, vol. 52, pp. 51:1–51:34, Jul. 2019.

[84] M. Nasri and M. Hamdi, "LTE QoS parameters prediction using multivariate linear regression algorithm," in *Proc. 22nd Conf. Innov. Clouds, Internet Netw. Workshops (ICIN)*, Feb. 2019, pp. 145–150.

[85] H. Zhang, D. Nettleton, and Z. Zhu, "Regression-enhanced random forests," *Section Stat. Learn. Data Sci.*, vol. abs/1904.10416, pp. 636–647, Jan. 2019.

[86] S. D. Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain," in *Proc. ITASEC*, 2018, pp. 1–11.

[87] J. B. Lasserre, "Sur l'inversion de matrice," *RAIRO-Oper. Res.*, vol. 26, no. 2, pp. 177–182, 1992.

[88] V. Eramo and F. G. Lavacca, "Optimizing the cloud resources, bandwidth and deployment costs in multi-providers network function virtualization environment," *IEEE Access*, vol. 7, pp. 46898–46916, 2019.

[89] *Digital Cellular Telecommunications System; Universal Mobile Telecommunications System (UMTS); LTE; Network Architecture*, document 3GPP TS 23.002 Version 11.6.0, Release 11, ETSI, 2013

[90] D. M. Manias, M. Jammal, H. Hawilo, A. Shami, P. Heidari, A. Larabi, and R. Brunner, "Machine learning for performance-aware virtual network function placement," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[91] L. J. Chaves, I. C. Garcia, and E. R. M. Madeira, "Ofswitch13: Enhancing Ns-3 with openflow 1.3 support," in *Proc. ACM Workshop Ns-3*, 2016, pp. 33–40.
[92] S. Panichpapiboon and W. Pattara-Atikom, "A review of information dissemination protocols for vehicular ad hoc networks," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 3, pp. 784–798, 3rd Quart., 2012.
[93] C. Vens and F. Costa, "Random forest based feature induction," in *Proc. IEEE 11th Int. Conf. Data Mining*, Dec. 2011, pp. 744–753.
[94] S. D. Angelis, "Assessing security and performances of consensus algorithms for permissioned blockchains," *CoRR*, vol. abs/1805.03490, May 2018.

SAIDA MAAROUFI (Senior Member, IEEE) received the M.Sc. degree in computer engineering from the Mohammed V University, Morocco, and the Ph.D. degree in computer engineering from the École Polytechnique de Montréal. She is currently a Postdoctoral Fellow with Polytechnique Montreal. Prior to that, she was a Network Engineer with DELL. Her research broadly revolves around wireless and mobile computing with specific interests on context-aware ubiquitous computing, interworking architecture design, machine learning, network virtualization and orchestration, and network security. She serves as the Chair for IEEE Montreal Section and the Vice-Chair for IEEE Women In Engineering (WIE) Canada. She was a recipient of several awards among which are the 2020 IEEE J. J. Archambault Canada Medal and the 2016 IEEE Canada Women in Engineering Prize sponsored by the Judy Clift Fund.

SAMUEL PIERRE (Senior Member, IEEE) received the B.Eng. degree in civil engineering from the École Polytechnique de Montréal, Montreal, QC, Canada, in 1981, the B.Sc. and M.Sc. degrees in mathematics and computer science from the Université du Québec a Montréal, Montreal, in 1984 and 1985, respectively, the M.Sc. degree in economics from the University of Montreal, Montreal, in 1987, and the Ph.D. degree in electrical engineering from the École Polytechnique de Montréal, in 1991. He is currently a Professor of Computer Engineering with the École Polytechnique de Montréal, where he is the Director of the Mobile Computing and Networking Research Laboratory and the NSERC/Ericsson Industrial Research Chair in next-generation mobile networking systems. He has authored or coauthored more than 500 technical publications, including articles in refereed archival journals, textbooks, patents, and book chapters. His research interests include mobile computing and networking, cloud computing, and electronic learning. He is a fellow of the Engineering Institute of Canada.