Received 30 November 2022, accepted 6 February 2023, date of publication 16 February 2023, date of current version 25 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3244886

RESEARCH ARTICLE

Internet of Intelligent Vehicles (IoIV): An Intelligent VANET Based Computing via Predictive Modeling

MUHAMMAD HARIS[1], MUNAM ALI SHAH[1], AND CARSTEN MAPLE[2], (Member, IEEE)
[1]Department of Computer Science, COMSATS University Islamabad, Islamabad 44550, Pakistan
[2]Secure Cyber Systems Research Group, Warwick Manufacturing Group (WMG), University of Warwick, CV4 7AL Coventry, U.K.

Corresponding author: Carsten Maple (CM@warwick.ac.uk)

This work was supported in part by the UKRI through Academic Centre of Excellence in Cyber Security Research - University of Warwick under Grant EP/R007195/1, in part by The Alan Turing Institute under Grant EP/N510129/1, in part by the PETRAS National Centre of Excellence for IoT Systems Cybersecurity under Grant EP/S035362/1, and in part by Autotrust under Grant EP/R029563/1.

ABSTRACT With the significant research and advancements in technologies, arose new applications such as autonomous driving and augmented/virtual reality. These applications required massive computational resources for the execution of various tasks. Utilizing vehicles resources in a distributed manner and collectively with the help of volunteer computing for various computational tasks is an emerging research area. The appropriate and intelligent decision in selecting a volunteer vehicle is crucial in this opportunistic network where information is exchanged between vehicles. In this paper, we propose Intelligent Volunteer Computing-based VANETs architecture to fulfill the computational requirements of vehicles applications intelligently. We propose selection criteria to select volunteers' vehicles capable of the execution of the computationally intensive task. In this study to rightly identify the volunteer vehicle for task execution, we use a machine learning approach that predicts the capability of certain vehicles in completing the task. Extensive experimentation is conducted for the prediction of the computing capability of optimal volunteer vehicles. We used nine different regression techniques on publicly available datasets. The results show these techniques can efficiently predict the capability of volunteers. By comparing the regression techniques, the results indicate that the ridge regression and support vector regression can significantly reduce the mean square error, relative absolute error, and root mean square errors. Simulations are conducted to compare the proposed scheme with the existing one.

INDEX TERMS Internet of Vehicles, machine learning, regression models, VANETs, volunteer computing.

I. INTRODUCTION

INTERNET of Intelligent Vehicles (IoIV) is one of the major enabling technologies for intelligent transportation systems (ITS). It can enable a wide range of intelligent applications, such as real-time navigation, dynamic road safety applications, autonomous driving, customer privacy, and auxiliary driving [1].

The increasing vehicular applications initiate the exponential rise in the demands of computational resources. In addition, due to the limited computing and storage resources, they are unable to accomplish efficient and real-time execution. Consequently, to assist these application's necessity, the concept of computation offloading arose where vehicles share a task with another vehicle for processing and execution [2].

In this study, the proposed Intelligent volunteer computing-based VANETs architecture (IVCBV) exploits volunteer computing in vehicles intelligently with help of machine learning approaches. When resources are underutilized in certain scenarios such as vehicles in congestion, traffic signals, parking lots, and smooth road. The vehicles resources can be jointly used for computation purposes. The machine learning techniques used predict the capability of vehicles to choose suitable volunteers for computation.

Volunteer computing is a distributed computing approach that allows users to share their idle computing resources to help execute computationally intensive tasks. The goal of volunteer computing is to provide reliable, low-cost, robust,

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

and scalable computing to users where large tasks are divided into smaller chunks for execution [3]. Specialized infrastructure is not required for volunteer computing. The rapid deployment and less economical computing infrastructure are highly encouraging to support future generation distributed computing systems. The significant advantage is to utilize the computing resources that are being wasted, which are relatively cheap computing infrastructures for the applications [4]. Therefore, volunteer computing can be applied for utilizing surplus resources in connected computing devices such as onboard units of vehicles. When vehicles are jammed in traffic or delayed in traffic signals for an extended period, accessing remote servers simultaneously by many vehicles causes remote servers for task offloading. The idea proposed in the Volunteer Computing Based VANETs (VCBV) is a new paradigm used for resource utilization and task execution in VANETs [5]. The main idea of the VCBV is to use the surplus resources of the vehicles and use them to perform joint computation tasks. Consider a scenario where there are 50 vehicles stopped at a traffic signal. Each vehicle is smart and can serve as a computing node. Similarly, if there are road closures, 100s of vehicles are stuck for hours. In this case, the surplus resources of these vehicles (such as CPU, RAM, and network antenna) can be used to perform computation tasks.

Edge computing is an architecture that allows substantial storage and computation resources at the proximity of the user edge [6]. It can meet the demands of delay-sensitive applications by reducing latency, and the size of data transfer through the network [7]. The vehicular edge computing (VEC) environment enables the computational offloading by offloading tasks to the VEC servers to acquire efficient computation that helps to increase the quality of service (QoS) for vehicular applications [8]. However, in comparison with cloud servers, VEC servers are resources more constrained by bounded resources due to the scalable deployment and economic factors. Therefore, the VEC server's resources are no longer sufficient to meet the needs of computationally intensive vehicular applications.

Vehicular Cloud Computing (VCC) is the promising paradigm that enables vehicles to share computational capabilities in the pattern of the cloud. The various vehicular application such as infotainment services, augmented reality, and self-driving car requires high-rate communication, low latency, and an enormous amount of computing resources [9]. However, Cloud servers, on the other hand, are located far away from vehicles, resulting in high latency and failing to meet the needs of these new promising applications. Subsequently, the provision of real-time service for these applications is challenging for VCC. Furthermore, offloading to remote clouds is not feasible for applications and services that are exclusively dependent on time, and location [10].

Fog computing, an extension of cloud computing, also provides intelligence and processing closer to the end-users. It reduces the latency and improves the response time of many applications. Vehicle fog computing (VFC) is also similar by integrating conventional vehicular ad hoc networks (VANETs) and fog computing. Fog nodes can be set up at the edge of the vehicular network to collect efficiently and process real-time traffic data and location-aware network responses. VFC facilitates a wide range of services and applications based on vehicles such as road safety, entertainment services, and intelligent traffic control systems [11]. Fog is preferable to the cloud by providing services with low latency to vehicles, computation, and storage. However, many open research issues exist, such as security, scalability, dynamic offloading, and a minimum number of fog node utilization [12].

This work is the extension of our previously published work [5] with the motivation to use machine learning techniques in VCBV. Decision-making through machine learning for appropriate volunteer selection in VCBV can enhance resource utilization and the task execution process.

The main contribution of the paper is summarized as

- An architecture called Intelligent Volunteer Computing based VANETs (IVCBV) is proposed by integrating VANETs, Volunteer Computing, and Machine learning.
- Demonstration of proposed architecture with selection criteria for volunteer vehicles selection.
- Implementation of nine regression techniques on the vehicular data to predict the computing capability of vehicles for computation.
- Comparative analysis of different regression techniques on multiple datasets is presented.

The remaining sections of the article are organized as follows. Section II provides related work in detail. The proposed IVCBV architecture is presented in Section III. Section IV presents the experimental analysis containing the details of the dataset, data preprocessing, implementation workflow, results, and analysis. Lastly, we conclude the paper in section V.

II. RELATED WORK

This section reviews the literature on computation offloading and resource-sharing schemes in various computing paradigms. We describe the literature on volunteer computing in VANETs. Furthermore, the different applications of VANETs in which machine learning techniques are used are reviewed.

With the advancement in communication systems and network technologies, the number of vehicles is increasing exponentially. The increased emerge new vehicular applications of high computational and storage needs of execution. The processing and storage resources available in a single vehicle are insufficient to execute the tasks on time and meet application requirements. Therefore, computational offloading is performed, which refers to migrating all or parts of an application to other processing units. These intensive tasks are shared entirely or partially with other resource-rich vehicles/remote servers for execution [13]. However, the concept of computation offloading is the decision to offload a task to another entity and the selection of entity [14].

MCC is a computing paradigm that combines the power of the cloud with the portability of mobile devices. It aims to improve the computational capabilities of mobile devices by leveraging the cloud's enormous resources [15]. The efficient scheme for resource scheduling and dynamic offloading is used to optimize the consumed latency and energy [16]. The goal of the work is to minimize the completion time and energy consumption of an application. The experimental results show that the proposed scheme performs efficiently over the existing schemes. However, task offloading to the remote cloud can affect performance, making the scheme unfavorable for delay-sensitive applications. In [17], Energy-Efficient Algorithm is proposed to balance the energy tradeoff in offloading decisions. The algorithm is based on Lyapunov optimization, which minimizes energy consumption by determining applications running on local, remote, and nearby cloudlets.

In VFC, resource allocation is also a vital challenge due to the geographic distribution of resources. Therefore, it's essential to allocate resources to reduce service latency properly. For such applications having various QoS requirements, the problem of admission control is tackled using the theoretical game method, which achieves scalability and QoS requirements [18]. The efficient offloading scheme is proposed to minimize latency and energy consumption. The selection policy is used to select a job for offloading from an overloaded cloudlet node while minimizing offloading costs and risks [19]. In [20], a methodology is presented for reducing base station load by utilizing unused vehicle resources through an effective incentive mechanism and a stable matching algorithm based on price. Parallel computing is an effective approach for completing tasks on time. A method of resource-aware-based parallel computing is used to select perfect nodes for task offloading [21]. Simulation is performed to validate the proposed method.

With the increasing amounts of vehicles, researchers started investigating resource sharing among vehicles. The idle resources of vehicles are utilized voluntarily for various applications needs. Resource-sharing strategies based on collaboration such as volunteer computing are becoming more popular to decrease repetitive work, speed up computing, and encourage the concept of decentralized open-source computing. In [22], a hybrid volunteer computing model in VANETs by utilizing spare resources of vehicles. The minimization of latency and maximization of system utility is taken into consideration. The coordination model utilizes resources for the execution of the task in a multi-hop fashion. Volunteer computing via VANETs was employed by Amjid et al. [23] to facilitate autonomous vehicles, with resources managed by a centralized job manager. The algorithms for different node registration methods were proposed to evaluate the job completion time and throughput. However, efficient resource utilization is not considered. Personal devices such as smartphones, tablets, and laptops execute computationally intensive tasks called personal volunteer computing. Studies show that personal devices can be leveraged for tasks such as crypto mining [24].

Besides the computational requirements, the VANETs applications need artificial intelligence-based techniques to improve the safety and intelligent transportation system [25]. Safety applications are types of VANETs application that focus on improving road safety. The major safety applications are pre-crash detection, U-turn assistance, collision avoidance, accident notification, and traffic sign violation. For preventing fatalities and traffic accidents, there is a need to implement safety applications on roads [26].

An intelligent crash prediction system using machine learning algorithms is used to predict crashes using vehicle data, including coordinates and speed of vehicles. The algorithms such as Random Forest (RF), Support Vector Machine (SVM), and Artificial neural network (ANN) are implemented on traffic data to identify normal and accident cases. The accuracy of Random Forest was higher than others. For determining road conditions, the support vector machine can be used because of its ability to give precise and timely information [26]. Convolutional Neural Networks (CNNs) are used to avoid risky moves to predict driver behavior and actions. The CNNs have higher accuracy and lower loss and efficiently predict traffic accidents [27]. The rapid growth and improvements in smart cities increase the number of challenging traffic issues. The distribution and storage of traffic sensor data by various individuals and organizations frequently lack trust, making it difficult for several parties to share sensor data. This creates a significant privacy risk. To deal with the issuing locality sensing hashing technique is proposed to predict traffic flow and protect privacy. The performance of the proposed technique is evaluated in terms of accuracy and efficiency [28]. On the basis of a real-world dataset with urban routes, an LSTM model-based speed forecasting method is presented [29]. The proposed models are created for univariate and multivariate scenarios and their accuracy for forecasting speed is evaluated. Simulation results demonstrate that for both short- and long-term forecasting, the multivariate model performs better than the univariate approach.

One of the possible uses of machine learning algorithms in the vehicular context is predicting the speed of vehicles and varying vehicular densities that frequently changes at different times of the day. The traffic management system based on an intelligent technique proposes reducing traffic issues like accidents, pollution, congestion, and wastage of fuel. The inputs for the proposed analysis and prediction model were historical data, speed, and density, and spatiotemporal correlation data [30]. The KNN technique is used to predict short-term traffic flow and describes the parameters that affect the port's short-term traffic flow as a state vector [31]. To predict future vehicle mobility for tens of minutes, RNN-based technique is used [32]. A theoretical analysis is performed to quantify the predictability of motion. The results indicate the quality of vehicle prediction is improved. For traffic flow

TABLE 1. Features that differentiate VEC, VCC, VCBV and proposed IVCBV.

Features	VEC	VCC	VCBV	IVCBV
Support of Mobility	✓	✓	✓	✓
Intelligent local decision making	✗	✗	✗	✓
Incentive supported	✗	✗	✓	✓
Improvement with time	✗	✗	✗	✓
Dynamic Selection	✗	✗	✗	✓
Network Maturity	✗	✗	✗	✓
Route Prediction	✗	✗	✗	✓
Applicable without internet	✓	✗	✓	✓
Performance Accuracy	✗	✓	✗	✓

prediction, four prediction models are used [33]. The four prediction approaches give extremely comparable results. Among them, the distributed random forest model slightly outperformed the other three.

In literature, a plethora of research has been done in numerous domains of volunteer computing, VANETs, and machine learning. However, these three areas are not explored jointly. In VANETs, the machine learning techniques are in various areas such as routing, classification of different networks, congestion detection, traffic prediction, etc. This research focuses on the usage of machine learning for volunteer computing in VANETs. The fundamental features that distinguish the VEC, VCC, VCBV, and the proposed IVCBV architecture are shown in Table.1.

III. IVCBV ARCHITECTURE

The Intelligent volunteer computing-based VANETs architecture is formed by integrating three conflated domains, namely volunteer computing, VANETs, and machine learning. The main focus is to utilize vehicular surplus resources intelligently and voluntarily for the needy vehicles with intensive tasks. The proposed architecture consists of task initiator vehicles, machine learning-enabled facilitator vehicles, and RSU, and lastly, the volunteer vehicles as shown in Figure 1. The initiator vehicle can be any vehicle looking for computation resources for the execution of the task. The task is the set of instructions that need resources for execution. The task varies because of the different needs of the vehicular application and the heterogeneity of a network. The non-delay sensitive applications such as infotainment services, weather forecasting, and vehicles assistance applications to improve quality of service are considered.

In VANETs, the unavailability of the internet makes it unsuitable for infrastructure based IVCBV. Vehicles communicate with each other using dedicated short-range communication (DSRC) called vehicle to vehicle communication (V2V). When vehicles are moving or stopped there OBU resources can be used for computation. In ad-hoc based there is no RSU so the vehicle can offload the computation task to volunteer vehicles using V2V. The machine learning-enabled facilitator vehicles are the central entity and act as a master vehicle between the initiators and volunteer vehicles. It receives a job from the initiator and schedules them to volunteer vehicles. It uses machine learning approaches on data provided by RSU and coordinates with both initiator vehicles and volunteer vehicles.

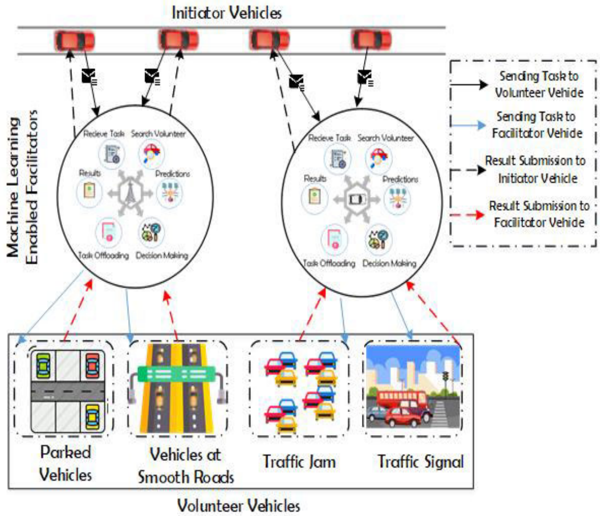

FIGURE 1. Proposed IVCBV architecture.

The communication between vehicles and infrastructure takes place which is commonly called as vehicle to infrastructure communication (V2I). The vehicles can communicate both with each other and with the RSUs through the OBUs. The OBUs have surplus resources which can be utilized for the computational requirements. Similarly, machine learning-enabled RSU also acts as a facilitator between the initiator vehicle and the volunteer vehicle. Through machine learning techniques, it can overcome numerous challenges. After the execution of the task by the volunteer vehicle, it collects results and submits them back to the initiator vehicles. The vehicles having surplus resources and willing to donate for the execution of the task with the collaboration of other vehicles are known as volunteer vehicles. These vehicles are in the communication range of initiator vehicles and facilitator vehicles at the time of task coordination and processing. These vehicles agreed to take part in the volunteer computing process. The nature of shared resources may vary due to the heterogeneity of vehicles, network conditions, and resource availability.

IVCBV considered multiple vehicles scenarios such as vehicles at a traffic signal, vehicles on smooth roads, and parked vehicles. When vehicles are waiting in traffic signals, the OBU resources can be collectively utilized for resource hunger vehicles. Although the waiting time is very short at a traffic signal, collectively, many vehicles can execute the small task in a short time and a distributed manner. Similarly, a large number of vehicles are parked in the parking lot for a long duration. Enough amount of pooled resources of parked vehicles can be sufficient for applications having higher demand, in smooth roads such as highways where violation of traffic rules and risk of accidents are less than busy inside city roads. Vehicles moving in the same direction for a long duration and can donate resources with each other. The machine learning-enabled facilitator vehicle behaves intelligently using machine learning approaches while coordinating

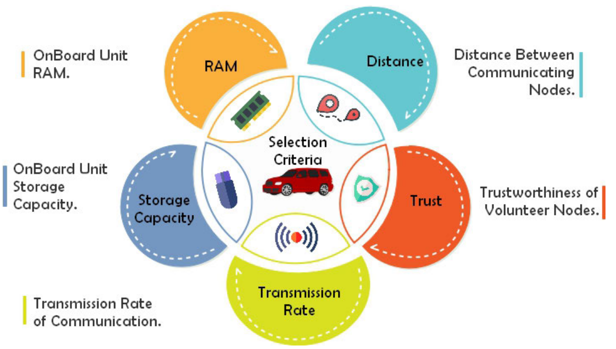

FIGURE 2. Features of selection criteria.

with other vehicles. It is probable to divide one task into various sub task and distribute it among many volunteers for executions. The main responsibility of the facilitator is to submit the correct results to the initiator vehicle on time without any delay and extra energy consumption.

A. IVCBV STEPS

The steps involved in IVCBV are as follow.

- Initiator vehicle sends the task to facilitator.
- Facilitator search volunteers in its premises to offload the task for execution.
- The facilitator use machine learning approaches for decision making and offload task to volunteers
- After completing execution the facilitator collects results from volunteers
- Final result submission to volunteer

B. SELECTION CRITERIA

It is essential to select the right volunteer's vehicle for task execution at the right time. The important concept of computational offloading is the decision of the selection of entities that perform the execution. After selection, the task is offloaded to the entity and starts executing. In our study, we defined a selection criterion for volunteer vehicles selection. The job facilitator will select the vehicle under dynamic selection criteria and offload the task to selected vehicles. The vehicular resource is frequently changing; therefore, vehicles with the highest resources are capable of the volunteer process each time. The features of selection criteria are presented in Figure 2. we have assumed that the selection criteria is based on the available resources. Certainly, we can think of incorporating more criteria to be selected as a volunteer node such as trustworthiness of a node to become volunteer; the time a volunteer vehicular node will remain available; the history of a node successfully completing jobs etc. Currently, the selection criterion selects the vehicle having the highest resources.

IV. EXPERIMENTAL ANALYSIS

In this section, we explain the datasets used for the prediction of the capability of vehicles. The data preprocessing and implementation workflow of regression techniques with results are discussed. The experimental setting of experimentation is provided.

TABLE 2. Summary of experimental details.

Parameters	Values
Regression Techniques	Linear Regression, K Nearest Neighbor, Support Vector Machine, Decision Tree, Random Forest, Gradient Boosting, XGBoosting, Adaboosting, and Ridge Regression
Environment	Google Colab
Programming language	Python (v3.7)
Train Test percentage	80,20
Visualization Tool	Flourish Studio
Dataset1 dimensions	7,18700
Dataset1 train size, test size	14960, 3640
Dataset2 dimensions	7,4600
Dataset2 train size, test size	3680, 920
Dataset3 dimensions	7,250
Dataset3 train size, test size	200,50
Evaluation Metrics used	R-Squared, Relative Absolute Error (RAE), Mean Absolute Error (MAE), Root Mean Square Error (RMSE).

Data collection and processing are basic and necessary for the training machine learning model in the machine learning pipeline. In this study, we used vehicular onboard unit computing capability dataset.[1] The datasets contain the capability based on different features. All of the datasets include vehicle id, RAM, storage capability, trust factor, distance, transmission rate, and eligibility score. The dataset1 contains seven features and 17800 instances. Dataset2 contains seven features and 4600 instances, and dataset3 contains seven features and 250 instances. The experimental setting are presented in Table 2.

A. DATA PREPROCESSING

Data preparation is one of the crucial steps in machine learning. It transforms the raw data into a useful and readable format before feeding it into the ML model. In our article, we used three different datasets of different sizes. To make data normally distributed, standardization is performed by using the sci-kit learn python library. The Standard Scaler is applied, standardizing the features by scaling to unit variance and removing the mean values.

B. IMPLEMENTATION WORKFLOW

Regression techniques are applied to each dataset separately. After preprocessing, the data is divided into two train and test sets. The regression techniques used are Linear Regression (LR), Support Vector Regression (SVR), K Nearest Neighbor Regression (KNN), Decision Tree Regression (DT), Random Forest Regression (RF), Gradient Boosting Regression (GB), XGBoosting Regression, AdaBoost Regression, and Ridge Regression. All the models are first trained on training data then test data is passed to evaluate the performance. The workflow is envisioned in Figure 3.

C. RESULTS

Four evaluation metrics are calculated to evaluate the performance of regression techniques. The prediction was

[1]Vehicles OBU Computation Capability (VOCC) https://data.mendeley.com/datasets/6fxbcs3hck/

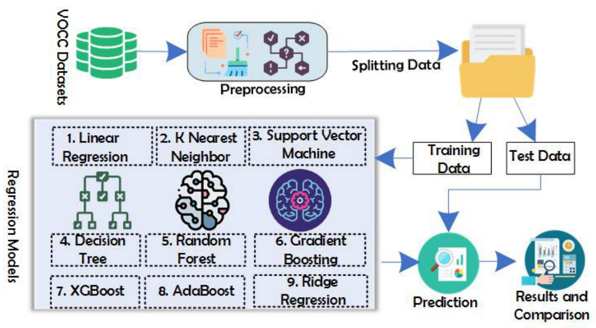

FIGURE 3. Machine learning workflow on VOCC datasets.

performed on multiple vehicular data to select the volunteer with the highest eligibility for computing certain tasks. The capable volunteer selection is necessary before assigning the task. The performance metrics to evaluate the regression techniques model are envisioned with discussion. This section represents the experimentation results of multiple datasets. Furthermore, the comparative analysis of multiple regression models is also presented.

R-Squared is the evaluation metric that evaluates the fitness of regression modelling. The higher r-square value indicates the fitness of the model. The maximum and optimal r-square values are 1. Mean Absolute Error is the evaluation metrics that are calculated by taking the mean of the absolute error. The error is the difference between the actual value and the predicted value for each instance. Relative Absolute Error is the ratio of mean residual error to naïve model error. A value close to zero is considered as good. Similarly, the Root Mean Squared Error is the standard deviation of predictive error and is considered an outstanding general-purpose error metric. RMSE is calculated by following steps. First taking the square of all errors, then take the mean of squared values, at the last step taking the square root of the mean value. The lower values represent the best fit of the model.

Figure 4, Figure 5, and Figure 6 shows the four metrics to evaluate the regression models on dataset1, dataset2, and dataset3 respectively. On dataset1 and dataset2, the r-squared of all models is above 90 per cent except for the Adaboost ensemble algorithm. The linear regressor, support vector regressor and ridge regressor give the highest and optimal value on dataset1, indicating these models best fit the volunteer data and identifying the relationship between the target variables and predictors. While for dataset3, the r squared value of most models is lower than 90. Comparatively, on all three datasets, the support vector regressor and ridge regressor give the lowest error and optimal value of RAE, MAE, and RMSE.

The data size of dataset3 is small, which affects the overall performance of models as shown in Figure 6. The prediction results of all metrics are comparatively low for dataset3. The decision tree and XGBoost model a having the lowest r-squared and highest error among all. The MAE and RMSE for KNN, RF, AdaBoost, and GB are above three as shown in Fig.6. Concluding, regression models have been applied to predict the capable volunteers in vehicular data. However, the support vector and ridge regression outperform and more accurately predict the volunteer eligibility score to select the right volunteer with the highest computing.

TABLE 3. Summary of simulation parameters.

Parameters	Values
Simulation software	Network Simulator 3.27
Number of volunteer	5, 10 ,15
Number of RSU	1
Data packet size	1024 B
Output packet size	150 B
Beacon frame size	40 B
PHY	IEEE802.11p
Transmission Protocol	UDP
Data rate	6 Mbps
Evaluation Criteria	Average Completion Time, Delay

The simulation experiments.[2] are also conducted to evaluate the proposed IVCBV model with the existing VCBV. The network simulator 3 is used for simulation [34]. It is a discrete and updated simulator for a network with rich features. It is an open-source software mostly used in research and education for communication models. Table 3 summarized the parameters of simulation performed for this study.

Wireless Access in Vehicular Environments (WAVE) is our experimental model's system architecture, which comprises one RSU and vehicular nodes. This architecture is utilized in the VANET environment, using standard IEEE 802.11p which is an extension of IEEE 802.11. This involves data transmission between vehicles as well as between RSU and vehicles. Some vehicles are uniformly deployed, with speeds ranging from 60 to 80 km/h, while some are in static position. As in our parked vehicle scenario the vehicles are in static position.

There are several assumptions for this study which are as follow

- If vehicle receives the hello beacon messages, then it is volunteer vehicle, and the vehicle is being used for volunteer computing.
- The physical layer parameter and channel condition are ideal.
- The hardware requirements for computation and communication are equipped in all vehicles.
- Jobs are compatible with the onboard unit and can divide into many tasks.
- Security parameters are configured and the network have no malicious vehicles.

A task is usually done by vehicle utilizing local computation but it can also offloaded to another vehicles having excess computation, to reduce job completion time. RSU is the centralized system that manages task scheduling. The job initiator is any vehicle. RSU is the job coordinator controlling the overall process, including initiation, task assignment, and output result gathering. The performance of IVCBV and VCBV is evaluated in terms of the time of execution and tasks. The average completion time of a different number of tasks is observed. Similarly, the completion time of tasks

[2]Code https://github.com/Haris-584/code

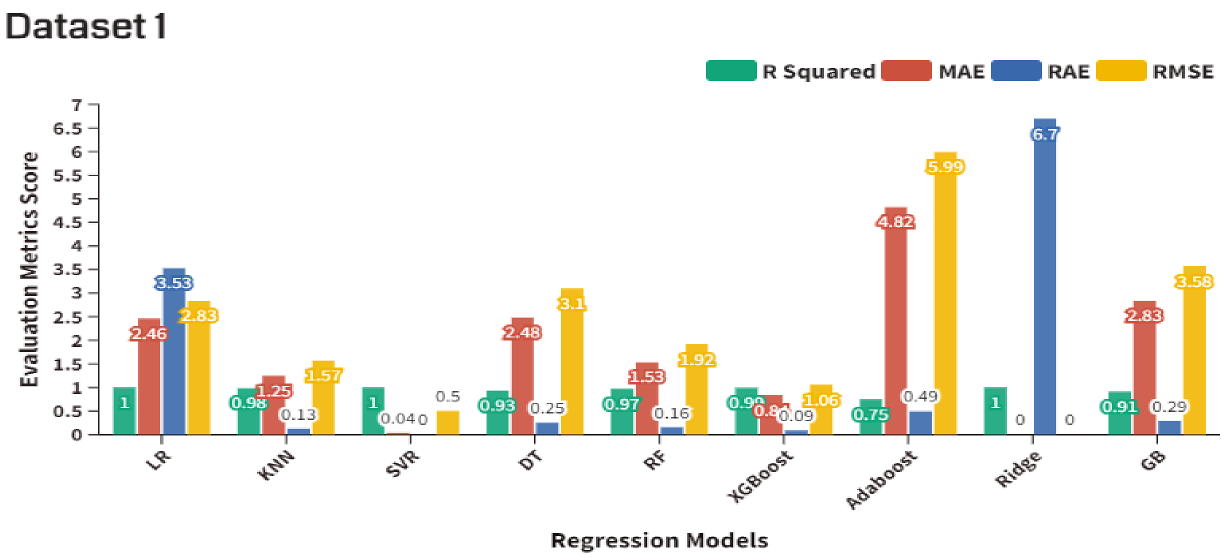

FIGURE 4. Performance of regression models on Dataset1.

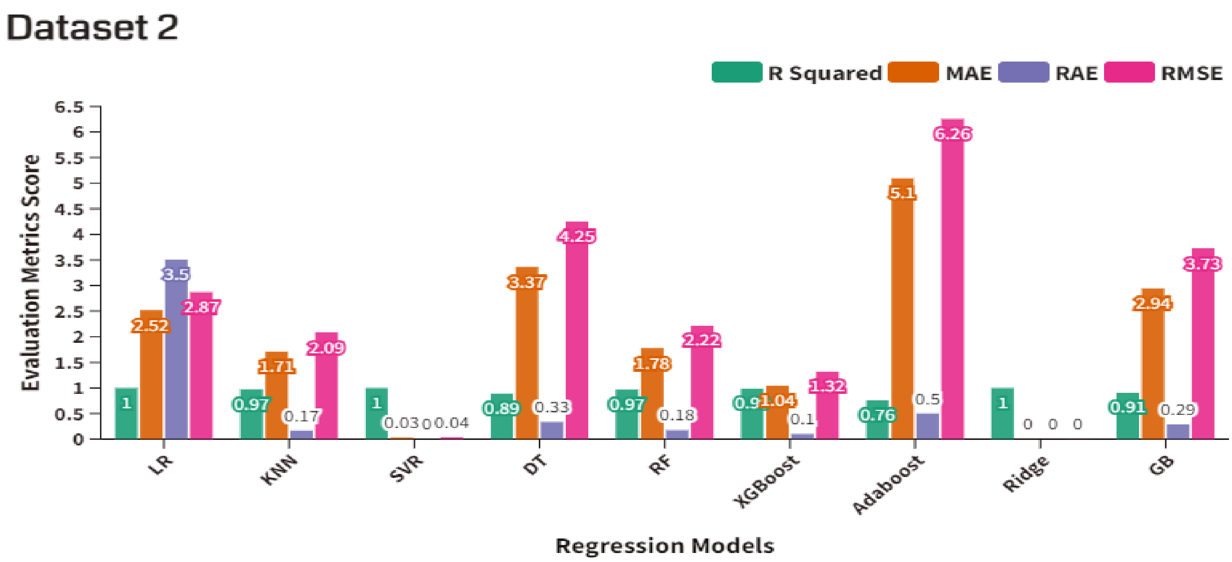

FIGURE 5. Performance of regression models on Dataset2.

having different sizes is also taken into consideration. The purpose of these performance parameters is to evaluate the task execution time and delay of the volunteer node for both systems.

Figure 7 shows the average execution time of a different number of tasks. In the scenario, a fixed task size of 1000 Kbits is set while the number is tasks are different. Initially, the IVCBV takes more time than VCBV because of the machine learning-based decision. Over time when the tasks increasing the task replication and task termination increases the VCBV average completion time. Noticeably, the IVCBV performs better than the existing one in which the replication and termination of the task are combated by assigning the task to the right volunteer. As the number of tasks increasing further there will be a major difference between both. In Fig. 7, we compare the number of tasks ranging from 5 to 40, Initially, the proposed IVBVC takes more time due to the training time taken by the machine learning algorithm, however, with an increasing number of tasks the execution time decreases, and the obvious reason is that now the proposed IVBVC has learned and gained the knowledge about previous network interactions. This is common with any scenario employing machine learning approach.

Figure 8 demonstrates the average execution time for a varied number of task sizes ranging from 300 to 1000 KB. It can be observed that for the smaller number of tasks the IVCBV takes higher time while for a higher number of tasks the performance of IVCBV is good as compared to the VCBV. The greater input size processing by incapable volunteer create retransmission which increases the time in

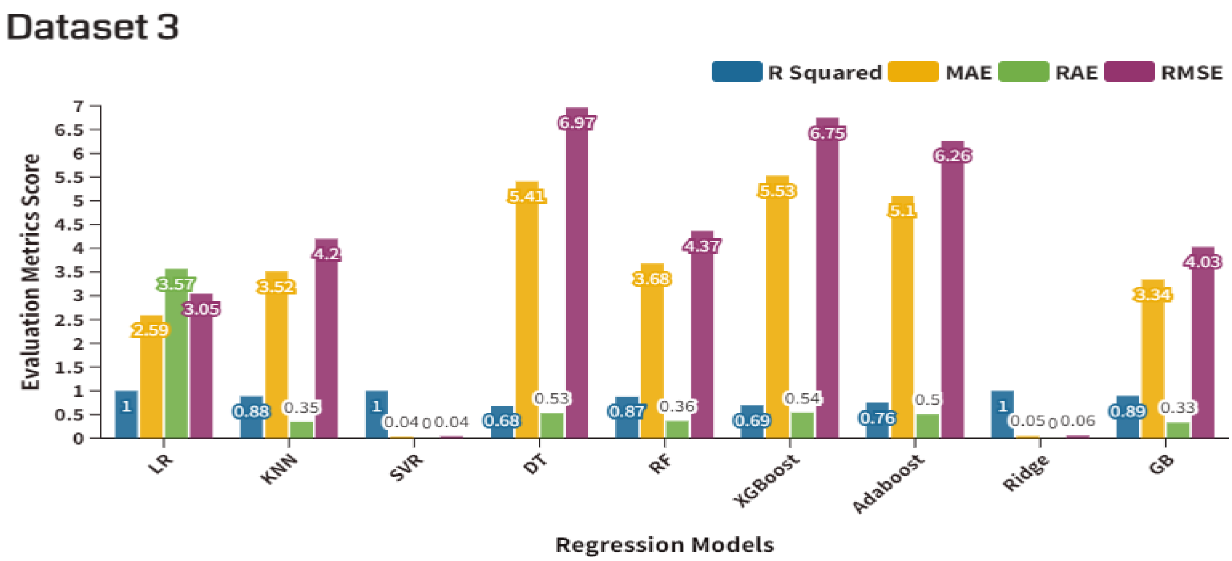

FIGURE 6. Performance of regression models on Dataset3.

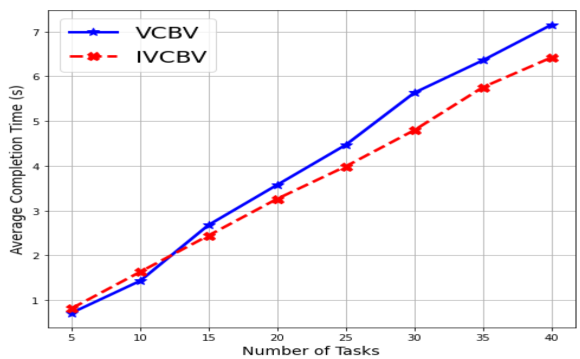

FIGURE 7. Comparison average completion time and number of tasks.

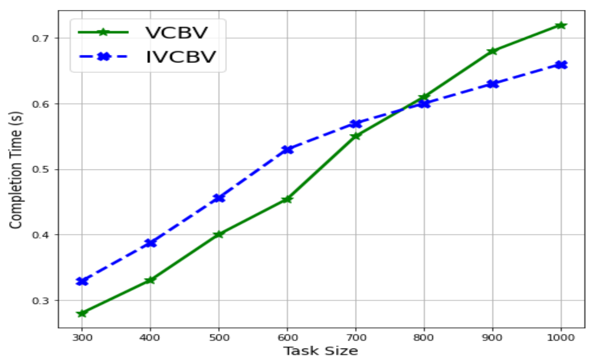

FIGURE 8. Comparision of completion time of different size of task.

VBCV. The reason for the proposed IVBVC taking more time for smaller tasks, smaller sizes, or smaller numbers of tasks is that the proposed IVCBV does not assign a task to a volunteer node randomly. We do not want a situation where a task is assigned to a volunteer vehicular node and later that node is unable to complete the job. In this case, we will have to look for another volunteer node and restart the same task. Obviously, this will take more time. It is better

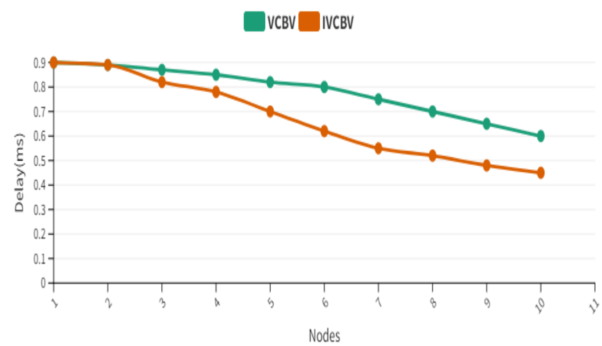

FIGURE 9. Comparison of delay and number of nodes.

to look for a suitable volunteer to assign the task which takes relatively more time, in the beginning, to find the suitable node. Opposite to this, in the existing VCBV, the task is directly assigned to a volunteer, therefore, the existing VCBV takes less time but later if the job is not completed, it will be reallocated resulting in more time taken.

Figure 9 illustrates the relationship between the end-to-end delay and the number of nodes. The packet delivery ratio of IVCBV slowly goes down as compared to VCBV. Initially, for a few nodes, the delay is same, with the increasing number of volunteer nodes the delay leads to a decrease. As the number of nodes rises, the time delay of both algorithms decreases, However, the IVCBV showed a downward trend to the increasing number of volunteer nodes and tasks.

The overall result is noticeably improved with time. There are two reasons for this attribute, one is the history of volunteers. When a volunteer executes a task, the historical data becomes available for a certain period. Next time, it is easier to assign a task to a particular volunteer. The results will surely improve with time. Another reason is that AI and ML

TABLE 4. Summary of results.

	R squared			MAE			RAE			RMSE		
	Dataset1	Dataset2	Dataset3	Dataset1	Dataset2	Dataset3	Dataset1	Dataset2	Dataset3	Dataset1	Dataset2	Dataset3
LR	1	1	1	2.4614	2.5202	2.5855	3.5289	3.5034	3.5696	2.8308	2.8687	3.048
KNN	0.983	0.9729	0.882	1.2508	1.7088	3.5155	0.1285	0.1697	0.3471	1.568	2.088	4.1971
SVR	**1**	**1**	**1**	**0.044**	**0.0318**	**0.0354**	**0.0045**	**0.0031**	**0.0035**	**0.501**	**0.0385**	**0.0449**
DT	0.9337	0.888	0.675	2.4817	3.3671	5.4105	0.2549	0.3344	0.53	3.0994	4.2495	6.9665
RF	0.9747	0.9695	0.8722	1.5325	1.7769	3.6804	0.1574	0.1765	0.3622	1.9169	2.218	4.3679
XGBoost	0.9922	0.9892	0.6948	0.8393	1.0409	5.5271	0.0862	0.1033	0.5435	1.06	1.3221	6.7508
Adaboost	0.7522	0.757	0.757	4.8245	5.0967	5.0967	0.493	0.5046	0.5046	5.9941	6.2576	6.2576
Ridge	**1**	**1**	**1**	**0.0006**	**0.0026**	**0.0482**	**6.6967**	**0.0002**	**0.0047**	**0.0008**	**0.0033**	**0.0577**
GB	0.9118	0.91	0.89	2.8335	2.9445	3.341	0.2911	0.2924	0.3325	3.5755	3.7309	4.0293

algorithms are deterministic, they learn and improve the result of training. The more training, the better the results.

Table 4 shows the summary of results of performance parameters of all regression models applied on three datasets. The results indicate that the SVR and Ridge regression outperforms and predicts the volunteer vehicle capability more accurately predicts the volunteers. The performance of dataset1 is comparatively better than others due to the training of models on a large number of data. Similarly, all the errors of dataset3 are high because of fewer data.

D. DISCUSSION

In the volunteer computing process, the task execution performs on idle devices which are not performing their actual job. However, if the device needs the resources for its primary role, this may be creating the termination of the already running task. If optimal volunteers are selected from the start when assigning the task, reduce the risk of task termination. Similarly, to avoid wrong results provided by volunteers, the task replication method can be used in which the same task assigns to multiple volunteers. A lot of energy can be wasted with task termination and task replication and is considered a major disadvantage. If the job facilitator predicts the optimal volunteer before offloading the task, it can overcome the numerous issues discussed above. Those volunteer vehicles having fewer resources will not be selected for the volunteer process to reduce the task termination and migration to other vehicles. Therefore, machine learning approaches significantly to resolve the selection of volunteers by predicting the capability of each vehicle. The evaluation of the regression technique indicates that overall results are extremely good with the least errors.

E. LIMITATION OF IVCBV

For delay-sensitive applications, the delay of task offloading, containing computing delay and transmission delay, is highly critical. IVCBV aims to entertain non time critical applications. However, the delay may occur in task offloading, machine learning algorithms, and task execution. Hence, minimizing the types of delay for different applications is also a challenge for IVCBV. Similarly, Machine learning can be used to analyze historic data to suggest future outcomes. The more and precise the past data the better results can be produced and can predict the future outcomes. whereas, in vehicles due to limited storage, and connectivity the previous data can be limited, and finite number of records can be return.

V. CONCLUSION

This study proposed a system that integrates three conflated domains: machine learning, volunteer computing, and VANETs. We illustrate the incorporation of machine learning for optimal decisions in the vehicles volunteer computing process. We used nine machine learning techniques to predict the computing capability of vehicles using publically available datasets. The techniques are validated through commonly used evaluation metrics. The results show that SVR and Ridge regression gives the most accurate results with the least errors. We hope that the proposed system and its premises will provide researchers with new opportunities to develop better solutions in this domain. In the future, we are intended to extend the work by examining the potential challenges like task selection for offloading, idle time prediction, and minimizing delay for delay-sensitive applications using machine learning techniques.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] A. Chowdhury, G. Karmakar, J. Kamruzzaman, and S. Islam, "Trustworthiness of self-driving vehicles for intelligent transportation systems in industry applications," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 961–970, Feb. 2020.

[2] F. Gu, X. Yang, X. Li, and H. Deng, "Computational resources allocation and vehicular application offloading in VEC networks," *Electronics*, vol. 11, no. 14, p. 2130, Jul. 2022.

[3] M. Nouman Durrani and J. A. Shamsi, "Volunteer computing: Requirements, challenges, and solutions," *J. Netw. Comput. Appl.*, vol. 39, pp. 369–380, Mar. 2014.

[4] T. M. Mengistu and D. Che, "Survey and taxonomy of volunteer computing," *ACM Comput. Surveys*, vol. 52, no. 3, pp. 1–35, Jul. 2019, doi: 10.1145/3320073.

[5] A. Waheed, M. A. Shah, A. Khan, S. U. Islam, S. Khan, C. Maple, and M. K. Khan, "Volunteer computing in connected vehicles: Opportunities and challenges," *IEEE Netw.*, vol. 34, no. 5, pp. 212–218, Sep. 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9183791/

[6] X. Xu, H. Li, W. Xu, Z. Liu, L. Yao, and F. Dai, "Artificial intelligence for edge service optimization in Internet of Vehicles: A survey," *Tsinghua Sci. Technol.*, vol. 27, no. 2, pp. 270–287, Apr. 2022.

[7] S. Chen, Q. Li, M. Zhou, and A. Abusorrah, "Recent advances in collaborative scheduling of computing tasks in an edge computing paradigm," *Sensors*, vol. 21, no. 3, p. 779, 2021.

[8] F. Zeng, Q. Chen, L. Meng, and J. Wu, "Volunteer assisted collaborative offloading and resource allocation in vehicular edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3247–3257, Jun. 2021.

[9] S. Das, A. R. Tripathy, and A. Tripathy, "Vehicular cloud computing: Architecture and its challenges," in *Smart Intelligent Computing and Applications*, vol. 2. Cham, Switzerland: Springer, 2022, pp. 481–493.

[10] Q. Huang, Y. Yang, and Y. Shi, "SmartVeh: Secure and efficient message access control and authentication for vehicular cloud computing," *Sensors*, vol. 18, no. 2, p. 666, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/2/666

[11] C. Huang, R. Lu, and K.-K. R. Choo, "Vehicular fog computing: Architecture, use case, and security and forensic challenges," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 105–111, Nov. 2017.

[12] M. A. Hoque and R. Hasan, "Towards an analysis of the architecture, security, and privacy issues in vehicular fog computing," in *Proc. SoutheastCon*, Apr. 2019, pp. 1–8.

[13] L. Zhao, K. Yang, Z. Tan, H. Song, A. Al-Dubai, A. Y. Zomaya, and X. Li, "Vehicular computation offloading for industrial mobile edge computing," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7871–7881, Nov. 2021.

[14] H. H. Elazhary, "Internet of Things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: Disambiguation and research directions," *J. Netw. Comput. Appl.*, vol. 128, pp. 105–140, Feb. 2019.

[15] Y. Liu and H. Song, "Levering mobile cloud computing for mobile big data analytics," in *Mobile Big Data*. Cham, Switzerland: Springer, 2018, pp. 21–39.

[16] S. Guo, J. Liu, Y. Yang, B. Xiao, and Z. Li, "Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 2, pp. 319–333, Feb. 2019.

[17] H. Wu, Y. Sun, and K. Wolter, "Energy-efficient decision making for mobile cloud offloading," *IEEE Trans. Cloud Comput.*, vol. 8, no. 2, pp. 570–584, Apr. 2020.

[18] J. Klaimi, S.-M. Senouci, and M.-A. Messous, "Theoretical game approach for mobile users resource management in a vehicular fog computing environment," in *Proc. 14th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2018, pp. 452–457.

[19] R. Yadav, W. Zhang, O. Kaiwartya, H. Song, and S. Yu, "Energy-latency tradeoff for dynamic computation offloading in vehicular fog computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 14198–14211, Dec. 2020.

[20] Z. Zhou, P. Liu, J. Feng, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Computation resource allocation and task assignment optimization in vehicular fog computing: A contract-matching approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3113–3125, Apr. 2019.

[21] J. Xie, Y. Jia, Z. Chen, Z. Nan, and L. Liang, "Efficient task completion for parallel offloading in vehicular fog computing," *China Commun.*, vol. 16, no. 11, pp. 42–55, Nov. 2019.

[22] A. Waheed, M. A. Shah, A. Khan, C. Maple, and I. Ullah, "Hybrid task coordination using multi-hop communication in volunteer computing-based VANETs," *Sensors*, vol. 21, no. 8, p. 2718, Apr. 2021. [Online]. Available: https://www.mdpi.com/1069556

[23] A. Amjid, A. Khan, and M. A. Shah, "VANET-based volunteer computing (VBVC): A computational paradigm for future autonomous vehicles," *IEEE Access*, vol. 8, pp. 71763–71774, 2020.

[24] E. Lavoie and L. Hendren, "Personal volunteer computing," in *Proc. 16th ACM Int. Conf. Comput. Frontiers*, Apr. 2019, pp. 240–246, doi: 10.1145/3310273.3322819.

[25] T. Li, C. Li, J. Luo, and L. Song, "Wireless recommendations for Internet of vehicles: Recent advances, challenges, and opportunities," *Intell. Converged Netw.*, vol. 1, no. 1, pp. 1–17, Jun. 2020.

[26] N. Dogru and A. Subasi, "Traffic accident detection using random forest classifier," in *Proc. 15th Learn. Technol. Conf. (LT)*, 2018, pp. 40–45. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8368509/

[27] H. Zhao, H. Cheng, T. Mao, and C. He, "Research on traffic accident prediction model based on convolutional neural networks in VANET," in *Proc. 2nd Int. Conf. Artif. Intell. Big Data (ICAIBD)*, May 2019, pp. 79–84.

[28] F. Wang, G. Li, Y. Wang, W. Rafique, M. R. Khosravi, G. Liu, Y. Liu, and L. Qi, "Privacy-aware traffic flow prediction based on multi-party sensor data with zero trust in smart city," *ACM Trans. Internet Technol.*, Mar. 2022, pp. 1–19.

[29] Y. N. Malek, M. Najib, M. Bakhouya, and M. Essaaidi, "Multivariate deep learning approach for electric vehicle speed forecasting," *Big Data Mining Anal.*, vol. 4, no. 1, pp. 56–64, Mar. 2021.

[30] S. Chavhan and P. Venkataram, "Prediction based traffic management in a metropolitan area," *J. Traffic Transp. Eng. English Edition*, vol. 7, no. 4, pp. 447–466, Aug. 2020.

[31] L. Yang, Q. Yang, Y. Li, and Y. Feng, "K-nearest neighbor model based short-term traffic flow prediction method," in *Proc. 18th Int. Symp. Distrib. Comput. Appl. Bus. Eng. Sci. (DCABES)*, Nov. 2019, pp. 27–30.

[32] W. Liu and Y. Shoji, "DeepVM: RNN-based vehicle mobility prediction to support intelligent vehicle applications," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 3997–4006, Jun. 2020.

[33] O. Mohammed and J. Kianfar, "A machine learning approach to short-term traffic flow prediction: A case study of interstate 64 in Missouri," in *Proc. IEEE Int. Smart Cities Conf. (ISC)*, Sep. 2018, pp. 1–7.

[34] *Network Simulator 3*. Accessed: Jan. 23, 2022. [Online]. Available: https://www.nsnam.org/

MUHAMMAD HARIS received the B.S. degree in computer science from the Kohat University of Science and Technology, Kohat, Pakistan, in 2018, and the M.S. degree in computer science from COMSATS University Islamabad, Pakistan. His research interests include artificial intelligence, machine learning, vehicular ad-hoc networks, and volunteer computing.

MUNAM ALI SHAH received the B.Sc. and M.Sc. degrees in computer science from the University of Peshawar, Pakistan, in 2001 and 2003, respectively, the M.S. degree in security technologies and applications from the University of Surrey, U.K., in 2010, and the Ph.D. degree from the University of Bedfordshire, U.K., in 2013. Since 2004, he has been working as an Assistant Professor with the Department of Computer Science, COMSATS University Islamabad, Pakistan. He has been included in Stanford list of top 2% scientists. He is leading the research group named as Performance Evaluation and Enhancements of Computing Systems (PEECS) within his department. He is a HEC approved supervisor and also supervised 3 Ph.D. students and 70 masters' students have completed their dissertation under his supervision. Currently, he is serving as a Tenured Associate Professor. He is the author of more than 250 research papers published in national and international conferences and journals. His research interests include Cyber Security, the Internet of Things, VANETs, Privacy Preservation, and Energy Efficient Communication Systems. He received the Best Paper Award from the International Conference on Automation and Computing, in 2012.

CARSTEN MAPLE (Member, IEEE) is the Principal Investigator of the NCSC-EPSRC Academic Centre of Excellence in Cyber Security Research at the University of Warwick and Professor of Cyber Systems Engineering. He is also a co-investigator of the PETRAS National Centre of Excellence for IoT Systems Cybersecurity, where he leads on Transport Mobility. He is a principal investigator on a $5 million project developing trustworthy national identity at the National Institute for Data Science and AI, U.K. He has an international research reputation, having published over 350 peer-reviewed papers, was a high-level scientific advisor to the European Commission, and given evidence to government committees on a variety of issues concerning safety, security, privacy and identity. He is a Fellow of the Alan Turing Institute, the National Institute for Data Science and AI, U.K.