Secure Data Dissemination Scheme for Digital Twin Empowered Vehicular Networks in Open RAN

Randhir Kumar, Prabhat Kumar, Ahamed Aljuhani, Alireza Jolfaei, A.K.M Najmul Islam and Nazeeruddin Mohammad

***Abstract*—The use of Open Radio Access Networks (Open RAN) in vehicular networks can lead to better connectivity, reliability, and performance. However, communication in this setting is often done over an unsecured wireless network, which creates a challenge in verifying the validity of received transactions by Internet of Vehicles (IoV) due to the untrusted network. It also creates a potential for attackers to tamper with the data content and conduct different IoV-related attacks. To address these issues, a new framework called "STIoV" has been proposed for secure and trustworthy communication in IoV. The framework includes a mutual authentication scheme to register and exchange session keys among the IoV participants, and a credit-based trust management system to assign reputation scores for the vehicular devices. The latter scheme discards transactions with low credit scores. To overcome the complexity and variability of the IoV network, digital twin technology is used to map Road Side Units (RSU) servers into virtual space, which facilitates constructing the vehicular relation model. An Intrusion Detection System (IDS) based on deep learning techniques is also introduced to detect anomalies in the traffic flow. The legitimate data is further used by the blockchain scheme for transaction verification, block creation and addition. Finally, the proposed framework has been evaluated based on two network intrusion datasets, and the results show the accuracy and efficacy of STIoV in comparison to several recent state-of-the-art solutions.**

***Index Terms*—Digital Twin, Deep Learning, Blockchain, Internet of Vehicles (IoV), Intrusion Detection System (IDS), Trustworthiness.**

I. Introduction

THE Internet of Things (IoT) has been embedded into many systems and deployed in a variety of critical sectors (e.g., transportation, vehicular, communications and energy) due to the numerous benefits that such technology offers. The Internet of Vehicles (IoV), a network of connected vehicle sensors, actuators, and smart devices that enables various objects to gather, send, and process data via the Internet, is the result of this evolution [1], [2]. The vehicular network is a critical component of IoV systems as it allows vehicles to communicate with other entities such as infrastructure, pedestrians, networks, grids, and cloud [3]. IoV networks require efficiency, high-speed connectivity, and reliable data transmission to achieve quality of service (QoS). However, the traditional radio access network (RAN) is incapable of supporting such a wide range of applications. Therefore, the open radio access networks (Open RAN) has emerged as an alternative approach for next-generation RAN [4]. As vehicular networks may have multiple vendors providing services in such a connected network, the Open RAN supports interoperability and interoperation in deployment between different vendors at a lower cost [5].

Although the Open RAN provides significant advantages to IoV companies and improves the quality of vehicular services for consumers, maintaining security and trust remains a considerable concern [6]. This is because the present Open RAN network mostly uses unsecured wireless communication channels to generate, collect, analyze and transmit vehicle-related data. This leads to various security and trust issues in the existing Open RAN network. These security issues mostly lead to cyberattacks that continuously pose a significant threat to IoV. The attackers use advanced methods and different techniques to disrupt vehicular services and cause tremendous damage. A cyberattack such as Distributed Denial of Service (DDoS) aims to disrupt vehicular services by rendering IoV unavailable or unresponsive to intended clients [7]. When these vehicular devices go out of service as a result of such an attack, it causes a significant impact on both service providers and consumers.

Generally, IoV systems use multiple sensors to enhance data reliability. But when observing the same object, whether from reliable, faulty, or corrupted sensors, the outcomes can vary. In addition, the truth must be deduced from the contradictory data, and any unreliable or insecure IoV nodes must be identified and addressed. As a result, ensuring trustworthiness across IoV nodes is critical in the development of a holistic vehicular system [8]. Trust management is categorized into three different domains namely distributed, semi-distributed, and blockchain-based. In distributed trust management, the trust is evaluated with the sensor capacity of disseminating data in the network. Thus, the success rate of trust is highly dependent on sensor capacity. The underlying approach is prone to bad-mouthing, Sybil, bad-collision, and re-entry attacks. In semi-distributed trust management, sensors share the information with trust value. The trust value is computed based on a threshold. However, the computation is completely done by a third party, thus, it is prone to a single point of

Randhir Kumar is with Department of Computer Science and Engineering, SRM University AP, India, AP 522240. (Email: randhir.honeywell@ieee.org).

Prabhat Kumar and A.K.M Najmul Islam are with the Department of Software Engineering, LUT School of Engineering Science, LUT University, Lappeenranta 53850, Finland (Email: prabhat.kumar@lut.fi, najmul.islam@lut.fi).

Ahamed Aljuhani is with the Department of Information Technology, University of Tabuk, Tabuk 71491, Saudi Arabia (e-mail:a_aljuhani@ut.edu.sa).

Alireza Jolfaei is with the Cyber Security and Networking College of Science and Engineering, Flinders University, Adelaide, Australia. (Email: alireza.jolfaei@flinders.edu.au).

Nazeeruddin Mohammad is with the Cybersecurity Center, Prince Mohammad Bin Fahd University, Alkhobar, Saudi Arabia (e-mail: nmohammad@pmu.edu.sa).

failure. This system is susceptible of various threats like man-in-middle, DoS, and DDoS [9]. The blockchain-based trust management consists of a policy-based mechanism or credit-based mechanism. The policy-based mechanism consists of Public Key Infrastructure (PKI) to maintain the trust in the network. The credit-based system computes the reputation score or credit score of entities and maintains the trust accordingly. The highest credit score assures more trustworthiness and vice-versa [10].

Vehicular vendors aim for efficient and secure IoV systems, but a persistent challenge in Open RAN-based IoV systems is ensuring data security and trust amidst diverse communicating entities, due to the varied and universal interconnected IoV network. This includes various communications like vehicle-to-network and vehicle-to-vehicle, which can modify vehicle records. With the advent of blockchain technology, several domains, including IoV, have benefited immensely from such a technology as it significantly transforms the vehicular ecosystem [11]. As blockchain technology provides a distributed and decentralized architecture, vehicular data can be securely processed, transferred, and shared with multiple distributed ledgers. This technology allows various stakeholders to store and exchange data and view vehicular records while ensuring the security and privacy of all transactions. Several existing studies using blockchain-based schemes for maintaining security and trust have been proposed in the literature, such as [12], [13]. However, many of these solutions lack a reliable, scalable, cost-effective, and trustworthy blockchain-based scheme [14], [15], [16]. The massive amount of heterogeneous and ambiguous data produced from IoV networks requires a cost-effective, efficient, and robust Intrusion Detection System (IDS) to overcome cyberattacks. Artificial intelligence (AI) plays a vital role in smart environments, and its application has been widely used in the IoV domain [17]. Machine Learning (ML) and Deep Learning (DL) have drawn a lot of interest when discussing AI because of their effectiveness in identifying malicious activity and minimizing system damage when combined with security features like the IDS. In this setting, DL models, such as recurrent neural networks and convolutional neural networks, excel at analyzing diverse and dynamic data patterns inherent in vehicular communication. By learning and adapting to evolving threats in real-time, these models enable the IDS to accurately identify anomalies and potential intrusions within the Open RAN, ensuring the security and integrity of communication in vehicular networks. Many existing studies such as [18], [19] used different ML and DL approaches in IoV environments; however, little work has been done toward blockchain-based schemes and DL as integrated frameworks in IoV networks.

Digital Twins (DTs) are an emerging digital mapping technology that can help capture socially aware healthcare services and improve the functioning of security and trust management systems by establishing a real-time digital simulation model of physical items [20]. DTs, in particular, implement in IoV a two-way closed-loop feedback mechanism of dynamic information. It not only gathers real-time operational data from IoV physical things, but it also implements control to modify the state of such entities [21]. In this article, we use DTs in IoV to map Road Side Units (RSUs) servers into virtual space, which facilitates constructing the vehicular relation model.

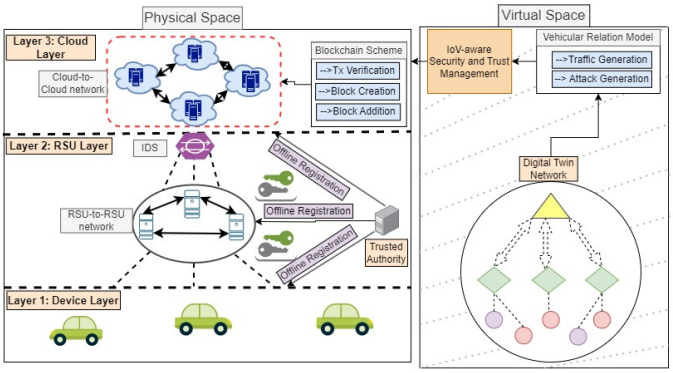


Fig. 1: Digital twin empowered IoV network

A. System Model

In this subsection, we discuss the system model consisting of a DT empowered IoV network and threat model that has been adopted and followed for this work.

1) Digital Twin Empowered IoV Network Model: The network model of STIoV mainly consists of two spaces referred to as physical and virtual space. Fig. 1 depicts the two-space architecture of the proposed STIoV framework. The description of these spaces is provided below.

- *Physical Space:* The physical layer consists of three layers: device layer, RSU layer, and cloud layer. The device layer primarily consists of vehicles that have limited computational power and are responsible for gathering traffic information and transmitting it on a hop-by-hop to RSU servers for processing and storage via a gateway. The RSU layer consists of relatively high compute-intensive and large storage space equipment deployed across geo-distributed sites, forming a peer-to-peer network. They are in charge of processing recent blockchain data, which includes a high number of recent vehicular transactions, in a short amount of time. Finally, there is a cloud layer, which is made up of numerous cloud platforms. The cloud servers in our model are distributed rather than being managed by a single entity. The incoming transactions from the RSU layer are initially screened by the DL-based IDS for intrusions. Finally, only normal transactions are forwarded to cloud servers. This layer builds a cloud-based peer-to-peer network of blockchain with a smart contracts facility. The smart contracts are used to validate the incoming transaction, execute a consensus algorithm with other cloud servers, and then add the block to the InterPlanetary File System (IPFS). The returned cryptographic hash is stored in the blockchain ledger.
- *Virtual Space:* The virtual space consists of the Digital Twin (DT) network and is designed on RSUs located in a particular area. It offers a vehicular relation model that is in charge of creating (for example, traffic and attack creation), transferring, and exchanging data across RSU

servers while keeping the consistency of their DT models. The model is produced, stored, and updated regularly based on the data gathered in the DT formation module. This virtual representation of the physical IoV network is initially used for training the proposed DL-based IDS. Once the IDS is trained, it can be deployed to screen real-time vehicular data or transactions.

2) Threat Model: The emergence of the vehicular network within Open Radio Access Networks (Open RAN) has brought forth a unique set of security challenges. Applying the Dolev-Yao (DY) threat model [22], we assume adversaries possess full control over the communication network. This means they can intercept, inject, modify, fabricate, and replay messages. In the context of IoV in Open RAN, such adversaries could eavesdrop on vehicle-to-vehicle or vehicle-to-infrastructure communications, manipulate traffic data or controls, and potentially disrupt the transport ecosystem. Conversely, the Canetti and Krawczyk (CK) model focuses on session-specific threats [23]. Here, adversaries target protocol executions or sessions. These adversaries can corrupt parties during sessions, violate confidentiality, and even hijack session keys. In the IoV scenario within Open RAN, this translates to threats like intercepting session establishments between vehicles and infrastructure, potentially derailing real-time communications vital for traffic management, safety mechanisms, and navigation aids. In the realm of IoV in Open RAN, both models underscore the importance of safeguarding communication integrity and the need for rigorous security measures.

B. Research Contribution

The following are the major contributions of this paper:

- We present a digital twin empowered Secure and Trustworthy communication framework called "STIoV" which comprehensively captures vehicular social features and improves security and trust in highly dynamic IoV networks.
- The STIoV provides an efficient mutual authentication and key agreement scheme between IoV, RSU, and cloud servers. The relevant and sensitive data are exchanged with a common session key over secure communication.
- A new blockchain-based credit-oriented trust management system for the IoV network is proposed, which uses blockchain technology to provide a reputation score to IoV for transparent trust evaluation.
- A novel DL-based IDS is also proposed to detect intrusions in the IoV network. The IDS consists of two main modules: first a "data pre-processing module" which combines data perturbation-driven encoding and normalization-driven scaling with an unsupervised generative DL architecture, Stacked Variational AutoEncoder (SVAE), for converting data into new format is proposed. Second a "multi-vector attack detection module" by using Attention-based Bidirectional Long Short-Term Memory (ABiLSTM) is proposed for detecting attack types in the IoV network.
- A novel blockchain-based transaction writing scheme is proposed. In this scheme, the authenticated and normal transactions collected by peer-to-peer cloud servers are used for transaction verification, block creation, and addition using a smart contract-based Proof-of-Authority (PoA) consensus algorithm. We also employed IPFS for transaction storage, and the resulting cryptographic hash is kept in the blockchain ledger, making it lighter and reducing block access time.

The remainder parts of this article are organized as follows. In Section II, the proposed framework's working and its core components are discussed. Experimental results by implementing DL and blockchain technology are presented in Section III. Finally, we conclude this paper with the future research directions in Section IV.

II. Proposed STIoV framework for Digital Twin Empowered Vehicular Networks

In this section, we have discussed the core components of our STIoV framework. This includes four core schemes, called "mutual authentication scheme", "Blockchain-based trust management scheme", "DL-based intrusion detection" and "blockchain-based transaction writing scheme". Each of these schemes is explained in detail below:

A. Proposed Mutual Authentication Scheme

1) Network Entities Initialization Phase: This section details the initialization process of network entities. The trusted authority (T_{AH}) registers all the entities of the network by executing certain parameters, which are mentioned below.

Step-1: The T_{AH} uses a non-singular elliptical curve $EL_{cv}(l, m)$ of form g^2= h^3+lh+m (mod r) over galois field *GF(r)*, where *r* is denoted as large prime number under condition of $4l^3$ + $27m^2$ $\neq$0 (mod r) for making it non-singularity as Ω for approximating zero point or infinity point. Further, T_{AH} chooses a base point *BP* $\in$ EL_{cv}(l,m) an order of closest over the *r* and *N*. That is, *N* . *BP* = Ω, where *N* . *BP* is denoted as a scalar multiplicative point over the elliptic curve and *N* $\in$ DS_r denoted as a discrete algorithm over a base point *BP*.

Step-2: The T_{AH} chooses a hash function over the principal of one-way message-digest hash for collision resilient, i.e., *HF(.)*. This is evaluated using a hash-based algorithm (SHA-256) for a reason of security, which produces 256-bit of unique message digest.

Step-3: T_{AH} picks an identity $ID_{T_{AH}}$ chooses a master key $MK_{T_{AH}}$ and creates a random private key $RPR_{T_{AH}} \in DS_r$, where DS_r = $\{1, 2, 3, 4, \ldots, r-1\}$. Next, T_{AH} finds public key $PB_{T_{AH}}$ = $RPR_{T_{AH}}$. *BP*.

Step-4: The T_{AH} preserves a $RPR_{T_{AH}}$, secret key $MK_{T_{AH}}$, and shares public parameters like $\{EL_{cv}(l, m)$, *BP*, $PB_{T_{AH}}$, *HF(.)*$\}$.

2) Registration of Entities: This section shows the registration process of all the required entities in the network i.e., cloud server, RSU server, and IoV.

Registration of Cloud Server: The T_{AH} makes registration of a cloud server (*CH*) and follows the steps mentioned below during its registration.

TABLE I: Authentication Process between IoV nodes and RSU Server

IoV nodes (IoV_i)	*RSU Server (FSV)*
Generate random number$mdr_1 \in DS_r$ uses a current timestamp $mCTS_1$ Compute mL_1= HF ($PSID_{IoV_i} \| TMID_{IoV_i} \| mdr_1 \| mCTS_1$) mL_2= $EN_{PB\,FS}(mL_1)$ mL_3= HF($mL_2 \| CRT_{IoV_i} \| PSID_{IoV_i} \| TMID_{IoV_i} \| mCTS_1$) MSG_1={$PSID_{IoV_i}$,$TMID_{IoV_i}$,$mCTS_1$,mL_2,mL_3} ⟶ (open channel)	
	Verify $\|mCTS_1^*$- $mCTS_1\| < \delta T$, if valid textitCRT_{IoV_i}.*BP*= $PB_{T_{AH}}$ + HF($PB_{IoV_i} \| PB_{T_{AH}}$), if validated successfully Fetch $PSID_{IoV_i}$ with respect to $TMID_{IoV_i}$ from the database Computes mL_3^*= HF ($mL_2 \| PSID_{IoV_i} \| TMID_{IoV_i} \| CRT_{IoV_i}$) Checks mL_3^*= mL_3, if validated successfully Decrypts mL_1= $D_{RPR_{FS}}(mL_2)$ Picks an unique random number $FSr_1 \in DS_r$ uses current timestamp $mCTS_2$ Computes FS_1= HF($PSID_{IoV_i} \| PSID_{FS} \| FSr_1 \| mCTS_2$) Encrypts FS_1 and stored in FS_2= $EN_{PB\,IoVi}(FS_1)$ produces a session key $SESS_{FS}$= HF($TMID_{IoV_i}^{new} \| mL_1 \| FS_1 \| mCTS_1 \| mCTS_2$), $TMID_{IoV_i}^*$= $TMID_{IoV_i}^{new} \oplus$ HF($PSID_{FS} \| TMID_{IoV_i} \| mCTS_2$), FS_3=HF($TMID_{IoV_i}^* \| FS_1 \| CRT_{FS} \| PSID_{FS} \| mCTS_2$) MSG_2=$TMID_{IoV_i}^*$,FS_2,FS_3,CRT_{FS},$PSID_{FS}$,$mCTS_2$ ⟵ (via open channel)
Checks $\| mCTS_2^*$ - $mCTS_2 \| < \delta T$ validated if, CRT_{FS}.*BP*=$PB_{T_{AH}}$ + HF($PB_{FS} \| PB_{T_{AH}}$) Decrypts FS_2 to receive FS_1= $D_{RPR_{IoV_i}}(FS_2)$ Evaluate FS_3^*= HF($TMID_{IoV_i}^* \| FS_1 \| CRT_{FS} \| PSID_{FS} \| mCTS_2$) if FS_3^*= FS_3 valid Computes $TMID_{IoV_i}^{new}$= $TMID_{IoV_i}^* \oplus$ HF($PSID_{FS} \| TMID_{IoV_i} \| mCTS_2$) Produces a session key $SESS_{IoV_i}$=HF($TMID_{IoV_i}^{new} \| mL_1 \| FS_1 \| mCTS_1 \| mCTS_2$) Produces current timestamp $mCTS_3$ Performs verification of session key $SESV_{IoV_i}$=h($SES_{IoV_i} \| CTS_3$) Change $TMID_{IoV_i}$ and $TMID_{IoV_i}^{new}$ in the database MSG_3=$SESSV_{IoV_i}$,$mCTS_3$ ⟶ (using open channel)	
	Verify $\| mCTS_3^*$ - $mCTS_3 \| < \delta T$ Verify $SESSV_{IoV_i}$= HF ($SESSV_{FS} \| mCTS_3$) if validated successfully Updates TID_{IoV_i} and $TID_{IoV_i}^{new}$ in the database securely. Both IoV_i and *FS* validates session key $SESSV_{IoV_i}$ (=$SESSV_{FS}$)

TABLE II: Symbol and Description

Symbol	Descriptions
T_{AH}	Third party
$MK_{T_{AH}}$, $RPR_{T_{AH}}$, $PB_{T_{AH}}$	Master key, Private key, and Public key of Third party
ID_{CH}, ID_{FS}, ID_{IoV_i}	Identity of Cloud server, Fog server, and IoV
$PSID_{CH}$, $PSID_{FS}$, $PSID_{IoV_i}$	Pseudo identity of Cloud server, Fog Server, and IoV
$RTST_{CH}$, $RTST_{FS}$, $RTST_{IoV_i}$	Registration Timestamp of Cloud server, Fog server, and IoV
$TMID_{CH}$, $TMID_{FS}$, $TMID_{IoV_i}$	Temporary Identity of Cloud server, Fog server, and IoV
CRT_{CH}, CRT_{FS}, CRT_{IoV_i}	Certificate of Cloud server, Fog Server, and IoV
RPR_{CH}, RPR_{FS}, RPR_{IoV_i}	Private key of Cloud server, Fog server, and IoV
PB_{CH}, PB_{FS}, PB_{IoV_i}	Public key of Cloud server, Fog server, and IoV
HF(.), $EL_{cv}(l, m)$, *BP*	Hash Function, Elliptical curve point l, and m, and Base point
$mCTS_1$, $mCTS_2$, $mCTS_3$	Current Timestamp of Cloud server, Fog server, and IoV
$SESV_{CH}$, $SESV_{FS}$, $SESV_{IoV_i}$	Session key of Cloud server, Fog server, and IoV

Step-1: The T_{AH} picks an unique identity of ID_{CH} and finds pseudo identity $PSID_{CH}$= HF($ID_{T_{AH}} \| MK_{T_{AH}} \| RTST_{CH}$), where $RTST_{CH}$ is registration timestamp of respective cloud server. Next, T_{AH} picks a temporal identity $TMID_{CH}$ and also picks a random secret key $RPR_{CH} \in DS_r$ and computes the public key by using PB_{CH} = RPR_{CH} . *BP*.

Step-2: The T_{AH} produces a certificate of *CH* as CRT_{CH} = $MK_{T_{AH}}$ + HF ($PB_{T_{AH}} \| PB_{CH} \| $) * $RPR_{T_{AH}}$ (mod r). Next, T_{AH} preserves a cloud credentials i.e., ($TMID_{CH}$, $PSID_{CH}$,CRT_{CH}, RPR_{CH}, $EL_{cv}(l, m)$, *HF(.)*) in memory and distributes a public key PB_{CH} as public.

Registration of RSU Server:

Step-1: The T_{AH} picks an unique identity of ID_{FS} and finds pseudo identity $PSID_{FS}$= HF($ID_{T_{AH}} \| MK_{T_{AH}} \| RTST_{FS}$), where $RTST_{FS}$ is registration timestamp of respective RSU server. Next, T_{AH} picks a temporal identity $TMID_{FS}$ and also picks a random secret key $RPR_{FS} \in DS_r$ and computes the public key by using PB_{FS} = RPR_{FS} . *BP*.

Step-2: The T_{AH} produces a certificate of *FS* as CRT_{FS} = $MK_{T_{AH}}$ + HF($PB_{T_{AH}} \| PB_{FS} \| $) * $RPR_{T_{AH}}$ (mod r). Next, T_{AH} preserves a RSU server credentials i.e.,

($TMID_{FS}$, $PSID_{FS}$,CRT_{FS}, RPR_{FS}, $EL_{cv}(l,m)$, *HF(.)*) in memory and distributes a public key PB_{FS} as public.

Registration of IoV:

Step-1: The T_{AH} picks an unique identity of ID_{IoV_i} and finds pseudo identity $PSID_{IoV_i}$= h($ID_{T_{AH}}$ || $MK_{T_{AH}}$ || $RTST_{IoV_i}$), where $RTST_{IoV_i}$ is registration timestamp of respective IoV. Next, T_{AH} picks a temporal identity $TMID_{IoV_i}$ and also picks a random secret key $RPR_{IoV_i} \in DS_r$ and computes the public key by using PB_{IoV_i} = RPR_{IoV_i} . ***BP***.

Step-2: The T_p produces a certificate of IoV_i as CRT_{IoV_i} = $MK_{T_{AH}}$ + HF ($PB_{T_{AH}}$ || PB_{IoV_i} ||) * $RPR_{T_{AH}}$ (mod r). Next, T_{AH} preserves a IoV credentials i.e., ($TMID_{IoV_i}$, $PSID_{IoV_i}$,CRT_{IoV_i}, RPR_{IoV_i}, $EL_{cv}(l,m)$, ***HF(.)***) in memory and distributes a public key PB_{IoV_i} as public.

3) Authentication Phase: This phase includes two different authentications in the network namely (i) authentication of IoV and RSU server and (ii) authentication of RSU server and cloud server. The authentication is based on session key approval from both entities. This mutual session key-based authentication ensures secure communication and data sharing in the network. These steps need to be completed to ensure session-based communication.

(i) IoV to RSU server Authentication

Step-1: IoV_i picks a unique random number $mdr_1 \in DS_r$ and uses a current timestamp $mCTS_1$ and finds mL_1= HF ($PSID_{IoV_i}$ || $TMID_{IoV_i}$ || mdr_1 || $mCTS_1$). Further, IoV_i encrypts mL_1 and stores in mL_2= $EN_{PB\cdot FS}(mL_1)$. Furthermore, IoV_i finds mL_3= HF(mL_2 || CRT_{IoV_i} || $PSID_{IoV_i}$ || $TMID_{IoV_i}$ || $mCTS_1$) and creates request message for access the channel MSG_1=$\{PSID_{IoV_i}$, $TMID_{IoV_i}$, $mCTS_1$, mL_2, $mL_3\}$ and share to RSU server via open channel.

Step-2: After receiving of MSG_1 by RSU server at time interval $mCTS_1^*$, RSU server verify the timestamp $|mCTS_1^*$- $mCTS_1| < \delta T$. After successful verification of timestamp, RSU servers enable certificate verification using CRT_{IoV_i}.***BP***= $PB_{T_{AH}}$ + HF(PB_{IoV_i} || $PB_{T_{AP}}$) if both timestamp and certificates are valid then RSU server extracts credential such as $PSID_{IoV_i}$ with respect to $TMID_{IoV_i}$ from database and finds mL_3^*= HF (mL_2 || $PSID_{IoV_i}$ || $TMID_{IoV_i}$ || CRT_{IoV_i}). Further RSU server verify mL_3^*= mL_3. If both computation matches, RSU server decrypts mL_2 and stored in mL_1= $D_{RPR_{FS}}(mL_2)$.

Step-3: The RSU server again picks an unique random number $FSr_1 \in DS_r$ and uses a current timestamp $mCTS_2$ and produces new temporary identity $TMID_{IoV_i}^{new}$ and evaluates mFS_1= HF($PSID_{IoV_i}$ || $PSID_{FS}$ || $mFSVr_1$ || $mCTS_2$) and encrypts FS_1 as FS_2= $EN_{PB\cdot IoV\cdot i}(FSr_1)$. Next, RSU server (*FS*) produces a session key $SESS_{FS}$= HF($TMID_{IoV_i}^{new}$ || mL_1 || FS_1 || $mCTS_1$ || $mCTS_2$), $TMID_{IoV_i}^*$= $TMID_{IoV_i}^{new}$ $\oplus$ HF($PSID_{FS}$ || $TMID_{IoV_i}$ ||$mCTS_2$), and FS_3=HF($TMID_{IoV_i}^*$ || FS_1 || CRT_{FS} || $PSID_{FS}$ || $mCTS_2$) and sends a reply message MSG_2= $\{TMID_{IoV_i}^*$, FS_2, FS_3, CRT_{FS}, $PSID_{FS}$, $mCTS_2\}$ and share to IoV_i through open channel.

Step-4: After receive of reply message (MSG_2) from the RSU server on certain time $mCTS_2^*$, IoV_i verifies whether $|$ $mCTS_2^*$ - $mCTS_2$ $| < \delta T$ is valid or not. If validated successfully, then IoV_i checks for certificate CRT_{FS}.***BP***=$PB_{T_{AH}}$ + HF(PB_{FS} || $PB_{T_{AH}}$). Further, IoV_i decrypts FS_2 to find FS_1= $D_{RPR_{IoV_i}}(FS_2)$. Furthermore, IoV_i evaluates FS_3^*= HF($TMID_{IoV_i}^*$ || FS_1 || CRT_{FS} || $PSID_{FS}$ || $mCTS_2$) and verify, if FS_3^*= FS_3 then IoV_i evaluates $TMID_{IoV_i}^{new}$= $TMID_{IoV_i}^*$ $\oplus$ HF($PSID_{FS}$ || $TMID_{IoV_i}$ || $mCTS_2$) and produces a session key $SESS_{IoV_i}$=HF($TMID_{IoV_i}^{new}$ || mL_1 || FS_1 || $mCTS_1$ || $mCTS_2$) and sends to *FS*. Further, IoV_i uses a current timestamp $mCTS_3$ and perform session key verification $SESSV_{IoV_i}$ by $SESSV_{IoV_i}$=HF($SESS_{IoV_i}$ ||$mCTS_3$) and changes the $TMID_{IoV_i}$ and $TMID_{IoV_i}^{new}$ in the database. Furthermore, IoV_i produces acknowledgement message as MSG_3=$\{SESSV_{IoV_i}$, $mCTS_3\}$ and shares to *FS* via open channel.

Step-5: After delivery of acknowledgment message MSG_3 at time interval $mCTS_3^*$, then *FS* checks timestamp with $|$ $mCTS_3^*$ - $mCTS_3$ $| < \delta T$ is valid or not. Next, *FS* checks $SESSV_{IoV_i}$= HF ($SESSV_{FS}$ || $mCTS_3$). After successful match, the *FS* establish a session key $SESSV_{IoV_i}$ (=$SESSV_{FS}$) with IoV_i. Finally, *FS* changes $TMID_{IoV_i}$ and $TMID_{IoV_i}^{new}$ in the database securely. Table. I illustrates the authentication process of IoV_i and *FS*.

(ii) Authentication of RSU and Cloud Server

Step-1: *FS* picks an unique random number $FSr_1 \in DS_r$ and uses a current timestamp $mCTS_1$ and finds LFS_1= HF (SID_{FS} || $TMID_{FS}$ || FSr_1 || $mCTS_1$). Further, FS encrypts LFS_1 and stored in LFS_2= $EN_{PB\cdot CH}$(LFS_1). Furthermore, FS evaluates LFS_3= HF(LFS_2 || CRT_{FS} || $PSID_{FS}$ || $TMID_{FS}$ || $mCTS_1$) and produces a request message MSG_1=$\{PSID_{FS}$, $TMID_{FS}$, $mCTS_1$, LFS_2, $LFS_3\}$ and shares with ***CH*** via open channel.

Step-2: After receiving of MSG_1 by cloud server at time interval $mCTS_1^*$, RSU server verifies the timestamp $|mCTS_1^*$- $mCTS_1| < \delta T$. After successful verification of timestamp, RSU servers enable certificate verification using CRT_{FS}.***BP***= $PB_{T_{AH}}$ + HF(PB_{FS} || $PB_{T_{AH}}$) if both timestamp and certificates are valid then RSU server extracts credential such as SID_{FS} with respect to $TMID_{FSV}$ from database and finds LFS_3^*= HF (LFS_2 || $PSID_{FS}$ || $TMID_{FS}$ || CRT_{FS}). Further, cloud verify LFS_3^*= LFS_3. If both computation matches, cloud server decrypts LFS_2 and stored in LFS_1= $D_{RPR_{CH}}$(LFS_2).

Step-3: The cloud server again picks an unique random number $CHr_1 \in DS_r$ and uses a current timestamp $mCTS_2$ and produces new temporary identity $TMID_{FS}^{new}$ and evaluates CH_1= HF($PSID_{FS}$ || $PSID_{CH}$ || CHr_1 || $mCTS_2$) and encrypts CH_1 and stored in CH_2= $EN_{PB\cdot FS}(CH_1)$. Next, cloud server (*CH*) produces a session key $SESS_{CH}$= HF($TMID_{FS}^{new}$ || LFS_1 || CH_1 || $mCTS_1$ || $mCTS_2$), $TMID_{FS}^*$= $TMID_{FS}^{new}$ $\oplus$ HF($PSID_{CH}$ || $TMID_{FS}$ ||$mCTS_2$), and CH_3= HF($TMID_{FS}^*$ || CH_1 || CRT_{CH} || $PSID_{CH}$ || $mCTS_2$) and sends a reply message MSG_2= $\{TMID_{FS}^*$, CH_2, CRT_{CH}, $PSID_{CH}$, $mCTS_2\}$ and share to FS via open channel.

Step-4: After receiving of reply message (MSG_2) from the cloud server on certain interval of time $mCTS_2^*$, FS verifies

TABLE III: Authentication Process between RSU Server and Cloud Server

RSU Server (FS)	*Cloud Server (CH)*
creates an unique random number$FSr_1 \in DS_r$ uses a current timestamp $mCTS_1$ Evaluates LFS_1= HF ($PSID_{FS}$ || $TMID_{FS}$ || FSr_1 || $mCTS_1$) LFS_2= $EN_{PB\cdot CH}$(LFS_1) LFS_3= HF(LFS_2 || CRT_{FS} || $PSID_{FS}$ || $TMID_{FS}$ || $mCTS_1$) $\xrightarrow{MSG_1=\{PSID_{FS},TMID_{FS},mCTS_1,LFS_2,LFS_3\}}$ (open channel)	
	Verify $\|mCTS_1^*$- $mCTS_1\| < \delta T$, if valid textitCRT_{FS}.*BP*= $PB_{T_{AH}}$ + HF(PB_{FS} || $PB_{T_{AH}}$), if valid Fetch $PSID_{FS}$ with respect to $TMID_{FS}$ from the database Computes LFS_3^*= HF (LFS_2 || $PSID_{FS}$ || $TMID_{FS}$ || CRT_{FS}) Verify LFS_3^*= LFS_3, if validated successfully Decrypts LFS_1= $D_{RPR_{CH}}$(LFS_2) Picks an unique random number $CHr_1 \in DS_r$ and uses a current timestamp $mCTS_2$ Computes CH_1= HF($PSID_{FS}$ || $PSID_{CH}$ || CHr_1 || $mCTS_2$) Encrypt CH_1 and stored in CH_2= $EN_{PB\cdot FS}(CH_1)$ produces a session key $SESS_{CH}$= HF($TMID_{FS}^{new}$ || LFS_1 || CH_1 || $mCTS_1$ || $mCTS_2$), $TMID_{FS}^*$= $TMID_{FS}^{new}$ $\oplus$ HF($PSID_{CH}$ || $TMID_{FS}$ ||$mCTS_2$), $mCSV_3$= HF($TMID_{FS}^*$ || CH_1 || CRT_{CH} || $PSID_{CH}$ || $mCTS_2$) $\xleftarrow{MSG_2=TMID_{FS}^*,CH_2,CRT_{CH},PSID_{CH},mCTS_2}$ (via open channel)
Verify | $mCTS_2^*$ - $mCTS_2$ | $< \delta T$ Verify if, CRT_{CH}.*BP*=$PB_{T_{AH}}$ + HF(PB_{CH} || $PB_{T_{AH}}$) Decrypts the CH_2 to receive CH_1= $D_{RPR_{FS}}(CH_2)$ Computes CH_3^*= HF($TMID_{FS}^*$ || CH_1 || CRT_{CH} || $PSID_{CH}$ || $mCTS_2$) if CH_3^*= CH_3 valid Computes $TMID_{FS}^{new}$= $TMID_{FS}^*$ $\oplus$ HF($PSID_{CH}$ || $TMID_{FS}$ || $mCTS_2$) Evaluates session key $SESS_{FS}$=H($TMID_{FS}^{new}$ || LFS_1 || CH_1 || $mCTS_1$ || $mCTS_2$) Uses current timestamp $mCTS_3$ Performs verification of session key $SESSV_{FS}$=h($SESS_{FS}$ ||$mCTS_3$) Change $TMID_{FS}$ and $TMID_{FS}^{new}$ in th database $\xrightarrow{MSG_3=SESSV_{FS},mCTS_3}$ (via open channel)	
	Verify | $mCTS_3^*$ - $mCTS_3$ | $< \delta T$ Verify $SESSV_{FS}$= HF ($SESSV_{CH}$ || $mCTS_3$) if validated successfully Change $TMID_{FS}$ and $TMID_{FS}^{new}$ in th database. Verify both session key FS and *CH* $SESSV_{FS}$ (=$SESSV_{CH}$)

whether | $mCTS_2^*$ - $mCTS_2$ | $< \delta T$ is valid or not. if it valid then FS checks for certificate CRT_{CH}.*BP*=$PB_{T_{AH}}$ + HF(PB_{CH} || $PB_{T_{AH}}$). Further, FS decrypts CH_2 to finds CH_1= $D_{RPR_{FS}}(CH_2)$. Furthermore, FS evaluates CH_3^*= HF($TMID_{FS}^*$ || CH_1 || CRT_{CH} || $PSID_{CH}$ || $mCTS_2$) and verify, if CH_3^*= CH_3 then FS evaluates $TMID_{FS}^{new}$= $TMID_{FS}^*$ $\oplus$ h($PSID_{CH}$ || $TMID_{FS}$ || $mCTS_2$) and produces a session key $SESS_{FS}$=HF($TMID_{FS}^{new}$ || LFS_1 || CH_1 || $mCTS_1$ || $mCTS_2$) and sends to *CH*. Furthermore, FS uses a current timestamp $mCTS_3$ and performs session key verification $SESSV_{FS}$ by $SESSV_{FS}$=HF($SESS_{FS}$ ||$mCTS_3$) and changes the $TMID_{FS}$ and $TMID_{FS}^{new}$ in the database. Furthermore, FS produces acknowledgment message as MSG_3=$\{SESSV_{FS}, mCTS_3\}$ and shares to *CH* via open channel.

Step-5: After delivery of acknowledgment message MSG_3 at time interval $mCTS_3^*$, then *CH* checks timestamp with | $mCTS_3^*$ - $mCTS_3$ | $< \delta T$ is valid or not. Further, *CH* checks $SESSV_{FS}$= HF ($SESSV_{CH}$ || $mCTS_3$). After successful match, the *CH* establish a session key $SESSV_{FS}$ (=$SESSV_{CH}$) to FS. Finally, *CH* changes $TMID_{FS}$ and $TMID_{FS}^{new}$ in the database securely. Table III illustrates the authentication process of FSV and CSV.

B. Proposed Blockchain-based Trust Management Scheme

The proposed model consists of three layers namely the IoV device layer, RSU layer, and cloud layer. The IoV device layer is responsible for generating the health data of patients. The correctness of the data must be checked before making the entire system reliable. These collected data from the IoV sensors can be noised, manipulated, and biased [24]. As the immutability feature of blockchain doesn't guarantee the risk of manipulation and malicious activities. The trust-based blockchain framework can assure security and can prevent malicious activities in the entire network. Moreover, the framework incorporates a digital twin empowered secure

Algorithm 1 IoV trust evaluation based on computation of $\mathcal{C}r$

procedure CREDIT_SCORE($\mathcal{T}v$)
  Set $\mathcal{T}r$=0
  $\mathcal{T}h$= range(min, max)
  Check $\mathcal{T}v$ with Dataset ($\mathcal{DS}$)
  /*Compute $\mathcal{C}r$ using $\mathcal{T}v$ and $\mathcal{T}h$ */
  if ($\mathcal{T}v$ = $\mathcal{T}h$) **then**
    $\mathcal{T}r$ +=1
  else
    $\mathcal{T}r$-=1
  end if
  /*Computation of Credit Score ($\mathcal{C}r$) */
  $\mathcal{C}r$ = (($\mathcal{T}r$)/10)/ $\mathcal{NTX}$), where $\mathcal{NTX}$ is no. of transactions.
  /* Check if $\mathcal{C}r$ for evaluation of trust */
  if ($\mathcal{C}r$ is valid) **then**
    "shared the transactions over the RSU nodes"
  else
    "Discard the transaction and keep observation over the IoV"
  end if
end procedure

Algorithm 2 SVAE procedure for encoding DT data

procedure **SVAE**(Unlabelled_dataset $\mathcal{D}$=$\{d_i\}_{i=1}^{\mathcal{N}}$, training_epochs = $\mathcal{K}$)
  Randomly Initialize parameters ϕ and θ
  for $epoch = 1 \rightarrow \mathcal{K}$ **do**
    for $i = 1 \rightarrow N$ **do**
      Draw N samples from dataset $\mathcal{D} = \{d_i\}_{i=1}^{\mathcal{N}}$
      Sample $\in$ from the noise distribution $\in$ $\sim N(0, 1)$
    end for
    Update gradients g according to Eq.1
    Use g to update parameters ϕ and θ
    Use g to update ϕ to update encoders weight matrix W_q
    Use g to update θ to update decoders weight matrix W_p
  end for
return An encoded dataset $\widehat{\mathcal{D}} = \{d_i\}_{i=1}^{\mathcal{T}}$ is obtained from the trained encoder network $q_\phi(v|d)$
end procedure

and trustworthy communication approach, capturing diverse vehicular social features to enhance security and trust in dynamic IoV networks. The system ensures fairness by implementing an efficient mutual authentication and key agreement scheme among IoV, RSU, and cloud servers, facilitating secure communication with a common session key. Moreover, the introduction of a blockchain-based credit-oriented trust management system transparently assigns reputation scores to IoV, minimizing biases and errors in trustworthiness evaluations within the network.

In this framework, we present a credit based trust system where the credit score is evaluated against the IoV. The highest credit score gets aligned with a high reputation in the network and vice versa. The trust over the collected data from IoV is computed via transaction patterns also known as transaction values. The transaction values are matched against the respective threshold also known as pattern. Based on transaction value ($\mathcal{T}v$) the transaction score ($\mathcal{T}r$) is computed if $\mathcal{T}v$ falls under the minimum or maximum range of threshold or the pattern. The range is matched against the real-time data collected over the IoV network scenarios. Further, $\mathcal{T}r$ is processed to compute a credit score ($\mathcal{C}r$) of a respective IoV. The high $\mathcal{C}r$ gets treated as more trustworthiness and vice versa. Based on the $\mathcal{C}r$ the transactions are shared over RSU nodes, otherwise, transactions get discarded from the respective device. The computation of credit score is shown in Algorithm 1.

C. Proposed Deep Learning Scheme for Intrusion Detection

The proposed DL-based IDS includes two modules; 1) First, a data pre-processing module with three key stages is designed and applied to the datasets. The underlying approach combines a data perturbation-driven encoding (i.e., label encoding technique) and normalization-driven scaling (i.e., min-max scaling technique) with an unsupervised generative DL architecture i.e., Stacked Variational AutoEncoder (SVAE). The data pre-processing module aims to first alter/convert and then learn hidden patterns of DT without knowing the actual class labels (i.e., anomalous or benign). 2) Second, the extracted and encoded/transformed data is fed to a multi-vector attack detection module i.e., Attention-based Bidirectional Long Short-Term Memory (ABiLSTM) for intrusion detection and identification. This approach therefore decreases the dependency on passive modes of threat detection that rely on the use of conventional models of intrusion detection (i.e., database of signatures or rules). Both modules working are explained below:

1) The Data Pre-processing module: In the data pre-processing module, first, a data perturbation-driven encoding technique is used in which the categorical variable is mapped to numeric values using a label encoding technique. Then, normalization-driven scaling is performed using the min-max scaling technique. This process helps in filtering and maintaining the analytical values of datasets. The details of both steps are discussed in [6]. Finally, by using Algorithm 2 the SVAE technique is applied for encoding data. Variational AutoEncoder (VAE) is a powerful probabilistic generative technique for learning representations of high-dimensional data. The VAE is trained by a set of adapted weights to encode the input dataset $\mathcal{D} = \{d_i\}_{i=1}^{\mathcal{N}}$, where d is the features and $\mathcal{N}$ is the records, exclusive of the class labels into a hidden representation. Next, the data codes are retrieved based on a set of generative weights derived from the latent representation of the data. Further, VAE assumes a latent variable v is used to generate the data $\mathcal{D}$. On v, we create a Gaussian prior $p_\theta(v) = N(v|0, 1)$, an encoder model $q_\phi(v|d)$ with ϕ as a parameter, and a decoder model $p_\theta(v|d)$ with θ as a parameter, where $N(0, 1)$ is the normal distribution with mean 0 and variance 1. Universal function approximators, such as neural network models [25], can be used for the encoder-decoder pair. Using stochastic gradient descent g to optimize the variational lower bound $\mathcal{G}$ on the marginal probability, the training dataset $\mathcal{D}$ is utilized to estimate the model parameters ϕ and θ:

$$\mathcal{G}\left(\theta, \phi; d^{(i)}\right) = -D_{KL}\left(q_\phi(v|d^{(i)}) \,||\, p_\theta(v)\right) + E_{q_\phi(v|d^{(i)})}\left[\log p_\theta\left(d^{(i)}|v\right)\right]. \quad (1)$$

Algorithm 3 The multi-vector attack detection from DT data

Input: Encoded Dataset $\widehat{\mathcal{D}} = \{d_i\}_{i=1}^{\mathcal{T}}$
Output: $\mathcal{O} = Normal \rightarrow 0$, $Threat_1 \rightarrow 1$, $Threat_2 \rightarrow 2$, $Threat_3 \rightarrow 3$ and so on.
Divide $\widehat{\mathcal{D}}$ into $\widehat{\mathcal{D}}^{Training}$ and $\widehat{\mathcal{D}}^{Testing}$ sets
$\widehat{\mathcal{D}}^{Training'}$ = pre-processing $\left(\widehat{\mathcal{D}}^{Training}\right)$
Add attention layer to BiLSTM using Eq. 14 to build ABiLSTM
for *each BiLSTM layer* $\mathcal{K} \in$ *BiLSTM neural network* **do**
    for *each cell* $\in$ *BiLSTM layer* $\mathcal{K}$ **do**
        $BiLSTM^{Tr_{Model}}$ = Train_*BiLSTM_Softmax*$\left(\widehat{\mathcal{D}}^{Training'}\right)$
    end for
end for
$\widehat{\mathcal{D}}^{Testing'}$ = pre-processing $\left(\widehat{\mathcal{D}}^{Testing}\right)$
while *True* **do**
    $Threat^{Prediction}$ = $BiLSTM^{Tr_{Model}}$ $\left(\widehat{\mathcal{D}}^{Testing'}\right)$
    $\mathcal{O}$ = $Threat^{Prediction}$
    return $\mathcal{O}$
end while

$-D_{KL}$ denotes Kullback-Leibler (KL) divergence in Eq.1 and can be analytically integrated since we conclude that both the previous $p_\theta(v)$ and the posterior approximation $q_\phi(v|d)$ are Gaussian. KL divergence is a regularizer that applies the prior $p_\theta(v)$ on the estimated posterior $q_\phi(v|d)$. As long as the batch size is large enough (e.g., 100), the $\log p_\theta\left(d^{(i)}|v\right)$ expectation, which corresponds to a reconstruction error, can be calculated with just one sample from $q_\phi(v|d^{(i)})$. We need a reparameterization in order to sample from $q_\phi(v|d^{(i)})$. We make a Gaussian isotropic statement such that

$$q_\phi(v|d^{(i)}) = N(v|\mu^{(i)}, \sigma^{2(i)} \times \mathcal{I}). \tag{2}$$

where $\mathcal{I}$ is the identity matrix. The encoder function output are $\mu^{(i)}$ and $\sigma^{(i)}$. A reparameterization $v = \mu + \sigma \odot \in$ can now be used, the element-wise product is referred to as $\odot$ and $\in \sim N(\in |0, 1)$ is an auxiliary noise variable i.e., $\in$. In addition, we developed a stacked VAE (SVAE), in which multiple VAE are cascaded, i.e., the previous layer's hidden layer output is used as the input of the following layer, and a layer-by-layer greedy training strategy is used.

2) The Multi-Vector Attack Detection Module: Given the encoded dataset $\widehat{\mathcal{D}} = \{d_i\}_{i=1}^{\mathcal{T}}$ i.e., $\widehat{D} = (d_1, \ldots, d_t)$, hidden vector sequence $h = (h_1, \ldots, h_t)$ and time step $t \in [1, t_f]$, for a specific time interval t_f, one vector of the input data $\widehat{\mathcal{D}}$ sequence is processed by Long-Short Term Memory (LSTM). Based on the three gate architectures, i.e., input, forget, and output gates, the LSTM architecture is built. The input gate allows the information to be processed without disturbance in each memory cell, and the output gate protects other units from irrelevant data disturbances [26]. As for forgetting units, it makes forgetting irrelevant information from memory. Finally, the forget unit allows the memory to forget irrelevant information. An LSTM network only considers the sequence's historical details and can capture the current state's dependency on the previous state (i.e., forward direction in context). However, it can be helpful for many sequence learning tasks to have access to both past and future contexts, especially for the DT sequence data in the actual prognostic applications [27]. The bidirectional LSTM (BiLSTM) at time t combines a forward LSTM $\rightarrow$ that uses past information at time $t-1$ and a reverse LSTM $\leftarrow$ that uses future information at time $t+1$ to process DT data. Algorithm 3 describes steps used by the BiLSTM-based attack detection module. The transition function is calculated using the below equations [28]:

$$\overrightarrow{h_t} = f\left(\overrightarrow{d_t}, \overrightarrow{h_{t-1}}; \overrightarrow{\Theta_{LSTM}}\right) \tag{3}$$

$$\overleftarrow{h_t} = f\left(\overleftarrow{d_t}, \overleftarrow{h_{t+1}}; \overleftarrow{\Theta_{LSTM}}\right) \tag{4}$$

The parameters $\overrightarrow{\Theta_{LSTM}}$ and $\overleftarrow{\Theta_{LSTM}}$ of BiLSTM are shared and learned during the time steps.

$$i_t = \sigma\left(\mathcal{W}_i \mathsf{d}_t + \mathcal{H}_i h_{t-1} + b_i\right), \tag{5}$$

$$f_t = \sigma\left(\mathcal{W}_f \mathsf{d}_t + \mathcal{H}_f h_{t-1} + b_f\right), \tag{6}$$

$$z_t = \tanh\left(\mathcal{W}_z \mathsf{d}_t + \mathcal{H}_z h_{t-1} + b_z\right), \tag{7}$$

$$c_t = z_t \odot i_t + c_{t-1} \odot f_t, \tag{8}$$

$$o_t = \sigma\left(\mathcal{W}_o \mathsf{d}_t + \mathcal{H}_o h_{t-1} + b_o\right), \tag{9}$$

$$h_t = \tanh\left(c_t\right) \odot o_t. \tag{10}$$

The variables $\mathcal{W}_{(.)}$ denotes the input weight matrix of backward pass and variables $\mathcal{W}_{(.)}$ of forward pass. The variable $\mathcal{H}_{(.)}$ and $\mathcal{H}_{(.)}$ indicates weights between two consecutive hidden states of forward and backward pass. The terms $b_{(.)}$ and $b_{(.)}$ represent the bias term. The symbols σ and $\tanh$ are sigmoid and tanh activation functions. The operator $\odot$ defines element-wise multiplication. The output $y(t)$ is calculated using the forward and backward functions.

$$y(t) = \sigma_y(\overrightarrow{h_t}, \overleftarrow{h_t}). \tag{11}$$

The function σ_y is capable of performing any of the four operations—concatenation, multiplication, addition, and averaging—and it concatenates the sequences of output neurons in hidden layers. To enhance the performance of attack detection, we incorporated an attention mechanism into our BiLSTM model. Unlike standard BiLSTM networks that use their most recent hidden state as output, a BiLSTM network with an attention mechanism multiplies the hidden states by trainable weights. The weight coefficient of the attention mechanism is computed as follows:

$$\mathcal{E}_\mathcal{T} = \tanh\left(\mathcal{W}_\mathcal{W} \mathcal{H}_\mathcal{T} + \mathcal{B}_\mathcal{W}\right) \tag{12}$$

$$\mathcal{A}_\mathcal{T} = \frac{\exp\left(\mathcal{E}_\mathcal{T}\right)}{\sum_{I=1}^{\mathcal{N}} \exp\left(\mathcal{E}_I\right)} \tag{13}$$

$$\mathcal{V}_\mathcal{T} = \sum_{t=1}^{N} \mathcal{E}_\mathcal{T} \mathcal{A}_\mathcal{T} \tag{14}$$

The essential characteristic of the BiLSTM layer's output vector $\mathcal{H}_\mathcal{T}$ at the $\mathcal{T}$-th time is represented by the following notations: $\mathcal{E}_\mathcal{T}$, $\mathcal{W}_\mathcal{W}$ denote weight matrices, $\mathcal{B}_\mathcal{W}$ signifies bias, and $\mathcal{A}_\mathcal{T}$ represents weighting factors. The output of attention at time $\mathcal{T}$ is expressed as $\mathcal{V}_\mathcal{T}$. An attention function with ReLU activation, implemented through a fully connected layer, processes the attention layer's outputs. Subsequently, the intrusion is determined by the softmax activation function, and the objective is assessed using categorical cross-entropy loss.

Algorithm 4 Proof-of-Authority based consensus mechanism to Verify and Create Block in Blockchain Network

State: $\mathcal{CH} \in ID_{\mathcal{CH}}$ List of miners,
$\mathcal{CB}_k = (\mathcal{AB}_k, \mathcal{FB}_k)$ $\mathcal{AB}_k$ denotes local blockchain whereas $\mathcal{FB}_k$ denotes DAG block
br $\rightarrow$ Records of block
pr $\rightarrow$ Parents node of block
mr $\rightarrow$ verifies block and transactions of block
sr $\rightarrow$ new block addition in chain
dr $\rightarrow$ time to validate record of block
function INITIATE()$_z$
  while Valid **do**
    sleep $\leftarrow$ CTS / dr, CTS $\rightarrow$ Timestamp
    if z $\in \mathcal{CSV}_l \wedge$ sr mod $|\mathcal{CSV}_l| ==$ z **then**
      $br.pr \leftarrow$ lblock($\mathcal{CB}_k$) , lblock $\rightarrow$ last block
      $br.CSV \leftarrow \mathcal{FB}_k$
      $br.sr \leftarrow$ sr
      $\mathcal{CB}_k \leftarrow (\mathcal{AB}_k \cup$ br, $\mathcal{FB}_k \cup br.pr$)
      disseminate ($\mathcal{CB}_k$)
      delay(dr)
    end if
  end while
end function
function CALULATESCORE($\mathcal{AB}_m$, $\mathcal{FB}_m$)
  returns UNIT512_MAX * depth($\mathcal{AB}_m$, $\mathcal{FB}_m$) - step-number($\mathcal{AB}_m$, $\mathcal{FB}_m$)
end function
function BLOCKDISSEMINATE($\mathcal{AB}_m$, $\mathcal{FB}_m$)
  if CalculateScore($\mathcal{AB}_m$, $\mathcal{FB}_m$) $>$ CalculateScore($\mathcal{AB}_k$, $\mathcal{FB}_k$) **then**
    CalculateScore($\mathcal{AB}_k$, $\mathcal{FB}_k$) $\leftarrow$ CalculateScore($\mathcal{AB}_m$, $\mathcal{FB}_m$)
  end if
end function
function ISSUCCESSFUL(br)$_z$
  Vote $\leftarrow \{\text{br}_{k.IoV} \mid \text{br}_k \in \mathcal{AB}_k \wedge \text{br}_{k.step} >= br.step \}$ return ($|Vote|$ * 2 $> |\mathcal{CSV}_l|$)
end function

D. Proposed Blockchain-based Transaction Writing Scheme

This phase includes verification and block creation over the proposed framework. A legitimate IoV can generate transactions in the network once it gets successfully validated using a mutual session-based key agreement. Next, the transaction gets validated using the PoA consensus mechanism, and the block gets created by the miners ($\mathcal{CSV}$) and disseminated in the network for verification. After successful verification by peer nodes (($\mathcal{CS}_l$)), the block gets appended into the blockchain network. The block contains $\mathcal{CB}_k = (\mathcal{AB}_k, \mathcal{FB}_k)$ $\mathcal{AB}_k$ denotes local blockchain whereas $\mathcal{FB}_k$ denotes DAG block. The block creation is done after the majority of voting (more than 50% of votes of peers). The details of block verification and creation are shown in Algorithm 4.

III. PERFORMANCE ANALYSIS

In the process of implementing simulations for the newly proposed STIoV (Secure Transactions in Internet of Vehicles) framework, a high-performance Tyrone personal computer was utilized. This computing system was equipped with dual Intel CPUs, each clocking in at 2.20GHz, and an impressive memory capacity of 128 GB RAM. For the deep learning (DL) methodologies employed within the study, we leveraged the capabilities of "keras", a specialized deep learning library that operates on top of the "TensorFlow" framework. Moreover, to effectively design the smart contract module integral to the research, we opted for the "Ethereum Rinkeby" test network – a well-recognized platform in the blockchain domain. Off-chain data storage requirements were facilitated using the InterPlanetary File System (IPFS), specifically the 0.4.19 version. The core objective was to evaluate the performance of the STIoV framework. To this end, two distinct IoT-based intrusion datasets were adopted: ToN-IoT ($\mathcal{D}_\alpha$) [29] and CICIDS-2017 ($\mathcal{D}_\beta$) [30]. Comprehensive details, characteristics, and structures of both datasets can be explored in the study [31]. Before feeding the data into our models, a significant preprocessing stage was conducted on both datasets. The methods and techniques for this preprocessing are detailed in [31]. After these preliminary steps, the datasets were systematically divided, allocating 70% for training purposes and the remaining 30% for testing to ensure a robust evaluation.

TABLE IV: Adopted SVAE parameters

Component	Configuration
Input Layer	Features: 44 from $\mathcal{D}_\alpha$ and 78 from $\mathcal{D}_\beta$ datasets
Encoder	Hidden Layers: 2 1) Layer 1: 128 nodes, tanh activation 2) Layer 2: 32 nodes, tanh activation
Decoder	Hidden Layers: 2 1) Layer 1: 32 nodes, tanh activation 2) Layer 2: 128 nodes, tanh activation
SVAE Model	Loss Function: 'kullback_leibler_divergence', Epochs: 10, Optimizer: 'adam', Batch Size: 50

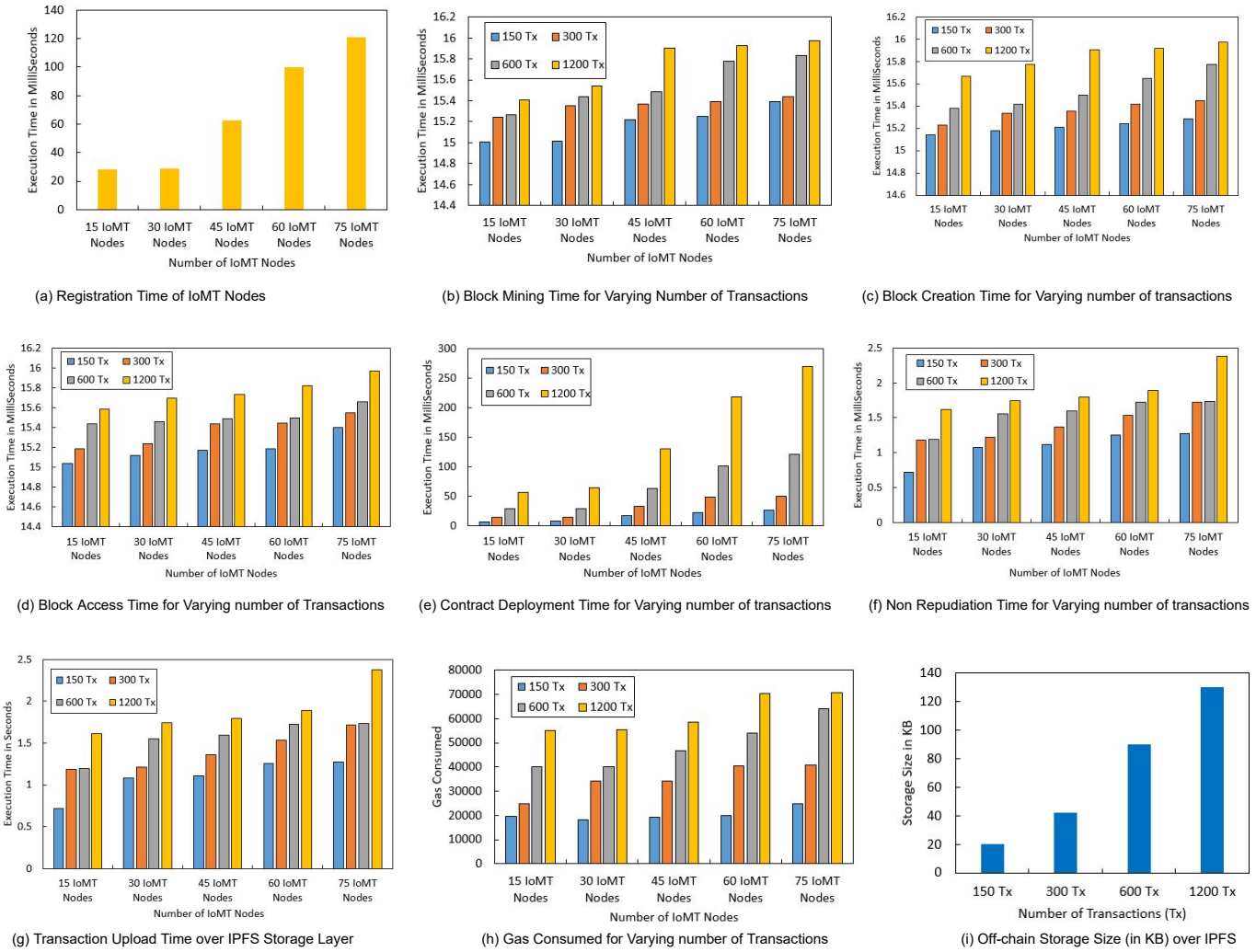
TABLE V: Adopted ABiLSTM-based attack detection parameters

Component	Configuration
Input Layer	Features encoded using SVAE technique
Hidden layers	Number of Layers: 5 1) Layer 1: 256 nodes, 2) Layer 2: 128 nodes, 3) Layer 3: 64 nodes, 4) Layer 4: 32 nodes, 5) Layer 5: 16 nodes, Dropout Rate: 0.2, Activation Function: relu.
Output layer	Units: 10 (1 normal and 9 attack for $\mathcal{D}_\alpha$), 11 (1 benign and 10 attack for $\mathcal{D}_\beta$), Activation Function: softmax
ABiLSTM Model	Loss Function: 'categorical_crossentropy', Optimizer: 'adam', Number of Epochs: 10, Batch Size: 50

A. Implementation results for blockchain-based schemes

The blockchain result analysis is shown in Fig. 2, where registration time, block mining time, and block creation time are computed. The registration time of IoV via the network is shown in Fig. 2(a), and it is clear that the registration time grows with the number of IoV. Figures 2(b), 2(c), and 2(d) illustrate block mining, block creation, and block access times for different numbers of peers and varying numbers of IoV. It is clear that when there are more devices or transactions

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2023.3342127



(a) Registration Time of IoMT Nodes

(b) Block Mining Time for Varying Number of Transactions

(c) Block Creation Time for Varying number of transactions

(d) Block Access Time for Varying number of Transactions

(e) Contract Deployment Time for Varying number of transactions

(f) Non Repudiation Time for Varying number of transactions

(g) Transaction Upload Time over IPFS Storage Layer

(h) Gas Consumed for Varying number of Transactions

(i) Off-chain Storage Size (in KB) over IPFS

Fig. 2: Blockchain Result Analysis

taking place on the network, each process's execution time lengthens. The contract deployment time is shown in Fig. 2(e). This procedure guarantees the timely completion of various smart contract processes. The non-repudiation time for various peer counts and transaction volumes is shown in Fig. 2(f). It is apparent that as there are more transactions in the network, the execution time is escalating significantly. The transaction upload time across the IPFS storage layer is depicted in Fig. 2(g). As the number of transactions rises, it is apparent that the execution time does as well. The time required to use gas for various numbers of networked transactions is shown in Fig. 2(h). The time required to consume gas is approximately the same for transactions with a low volume, but increases for transactions with a high volume. The amount of the transaction off-chain storage in KB is shown in Fig. 2(i). It is clear that the execution time and storage rely on the quantity of transactions uploaded to the safe storage layer of IPFS.

B. Implementation results for DL-based IDS

A data perturbation-driven encoding and normalization-driven scaling with a DL-based SVAE technique is designed and applied in the data pre-processing module. This method filters, and encodes data into a new format, and retains the analytical values of both datasets. The adopted hyper-parameters for executing SVAE are illustrated in Table IV.

The proposed data pre-processing module efficiency is evaluated by designing a utility system i.e., an ABiLSTM-based attack detection module. The performance is compared in two scenarios, i.e., with actual (i.e., before applying data pre-processing module) and transformed (i.e., after applying data pre-processing module) $\mathcal{D}_\alpha$ [29], and $\mathcal{D}_\beta$ [30] datasets. We have evaluated the results based on a multi-vector attack scenario for both datasets. The evaluation metrics such as accuracy vs loss, class-wise prediction, and confusion matrix are used to evaluate the performance of the proposed framework. Table V shows the adopted hyper-parameters in designing an ABiLSTM-based attack detection module.

1) The Accuracy vs Loss metric: First, we have used the accuracy vs loss evaluation metric to measure performance. Fig 3 and Fig 4 illustrate accuracy vs loss ABiLSTM-based attack detection module, after and before applying data pre-processing module using $\mathcal{D}_\alpha$ and $\mathcal{D}_\beta$ datasets. The accuracy vs loss for the ABiLSTM-based attack detection module using

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2023.3342127

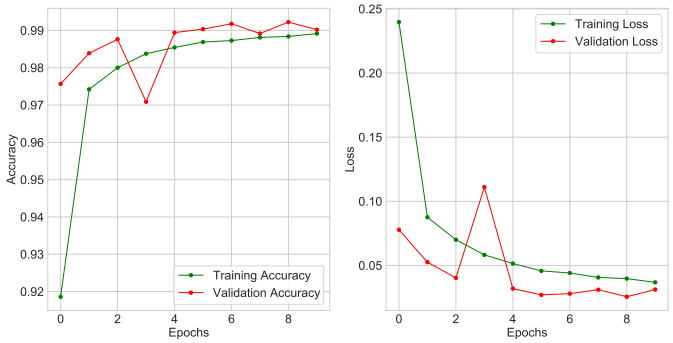


Fig. 3: The Accuracy vs loss for ABiLSTM-based attack detection module with transformed $\mathcal{D}_\alpha$ dataset

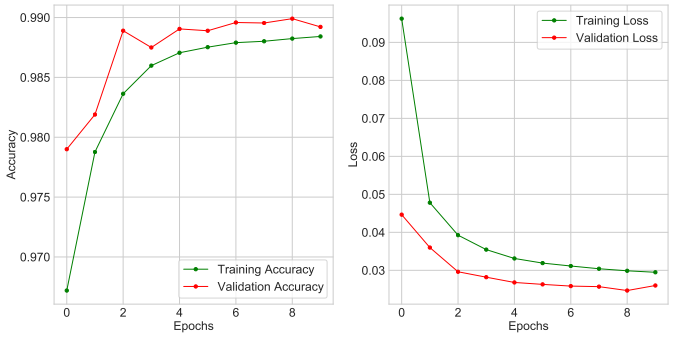


Fig. 4: The Accuracy vs loss for ABiLSTM-based attack detection module with transformed $\mathcal{D}_\beta$ dataset

the actual $\mathcal{D}_\alpha$ dataset is 99.07% and 0.0388% respectively, whereas the accuracy vs loss for the modified dataset is 99.09% and 0.0312%. Similarly, with the actual $\mathcal{D}_\beta$ dataset, the obtained values are 98.73% and 0.0285%, respectively, whereas with the modified dataset, they are 98.49% and 0.0260%.

2) Confusion Matrix: The Confusion Matrix (CM) summarizes the number of records that the proposed scheme detects correctly or incorrectly. The Overall CM is formed based on all attack and normal classes. Misclassified vectors are represented by off-diagonal elements in CM. The instances in a predicted class are displayed in each column, and the instances in an actual class are displayed in each row inside the CM. In a multi-attack scenario, Fig 5 shows the CM obtained using transformed $\mathcal{D}_\alpha$ and Fig 6 shows the CM obtained using transformed $\mathcal{D}_\beta$ datasets. The CM reveals that performance based on $\mathcal{D}_\alpha$ dataset is outstanding. However, for two attack groups (i.e., Bot and Web attack) of $\mathcal{D}_\beta$ dataset, the attack detection module has not performed well. This is due to the lower training instances present in the dataset. Overall, for both datasets ABiLSTM-based attack detection module has shown remarkable results.

C. Comparative analysis with baseline algorithms

Based on two datasets i.e., $\mathcal{D}_\alpha$ and $\mathcal{D}_\beta$, the performance of ABiLSTM-based attack detection module is compared to peer ML methods such as Decision Tree (DT), Random Forest (RF), and Naive Bayes (NB) in multi-class attack detection.

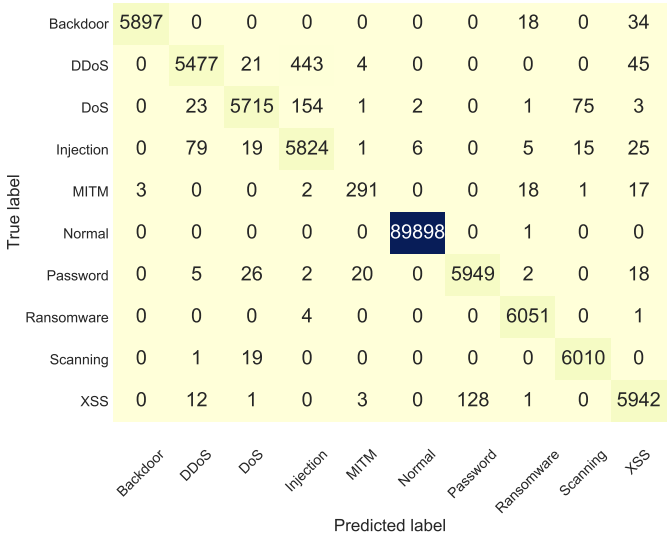


Fig. 5: Confusion matrix with transformed $\mathcal{D}_\alpha$ dataset

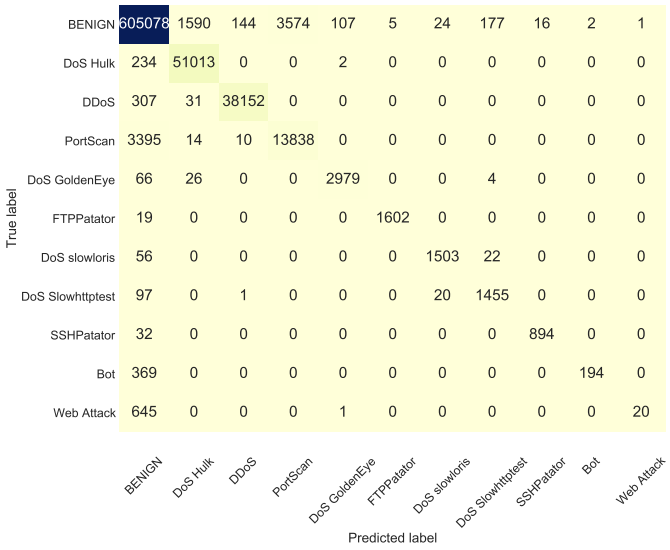


Fig. 6: Confusion matrix with transformed $\mathcal{D}_\beta$ dataset

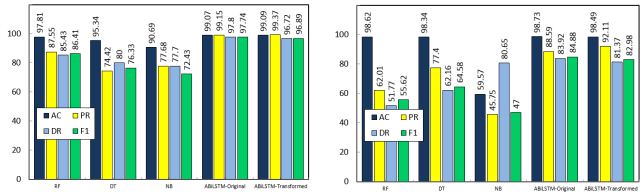
Table VI and VII illustrate multi-class DR comparison with peer ML techniques. It shows that for the majority of the attack vectors present in $\mathcal{D}_\alpha$ dataset the DR is on an average between $91\% - 100\%$. The ABiLSTM-based attack detection module has not done well for Bot and Web attacks when utilizing the $\mathcal{D}_\beta$ dataset, though. This is a result of the dataset's lower instance count. Fig 7a and Fig 7b shows AC, PR, DR and F1 values obtained by ABiLSTM-based attack detection module as 99.07%, 99.15%, 97.8%, 97.74% with actual and 99.09%, 99.37%, 96.72%, 96.89% with transformed $\mathcal{D}_\alpha$, and 98.73%, 88.59%, 83.92%, 84.88% with actual and 98.49%, 92.11%, 81.37%, 82.98% with transformed $\mathcal{D}_\beta$ datasets respectively. The proposed ABiLSTM-based attack detection module clearly outperforms the existing baseline schemes, as evidenced by the results.

TABLE VI: Multi-vector DR (%) comparison against baseline algorithms based on $\mathcal{D}_\alpha$ dataset

Techniques	Backdoor	DDoS	DoS	Injection	MITM	Normal	Password	Ransomware	Scanning	XSS
DT	100.00	100.00	100.00	0.00	0.00	100.00	100.00	100.00	100.00	100.00
RF	99.98	90.40	91.97	93.53	0.00	100.00	97.81	99.40	95.74	85.47
NB	99.22	26.80	91.70	92.96	95.11	100.00	75.32	79.98	96.91	19.02
ABiLSTM (with Actual)	99.73	93.95	99.34	93.82	99.39	100.00	97.62	99.98	97.01	97.20
ABiLSTM (with Transformed)	99.12	91.43	95.66	97.48	87.65	99.99	98.78	99.91	99.66	97.61

TABLE VII: Multi-vector DR (%) comparison against baseline algorithms based on $\mathcal{D}_\beta$ dataset

Techniques	BENIGN	DoS Hulk	DDoS	PortScan	DoS GoldenEye	FTPPatator	DoS slowloris	DoS Slowhttptes	SSHPatator	Bot	Web Attack
DT	100.00	90.00	99.00	97.00	66.00	99.00	35.00	0.00	97.00	0.00	0.00
RF	100.00	95.00	100.00	97.00	50.00	72.00	0.00	55.00	0.00	0.00	0.00
NB	55.00	89.00	98.00	50.00	99.00	100.00	60.00	77.00	97.00	76.00	08.00
ABiLSTM (with Actual)	99.77	95.54	98.47	77.51	93.85	99.56	96.90	98.15	96.54	64.65	02.55
ABiLSTM (with Transformed)	99.07	99.53	99.12	80.18	96.87	98.82	95.06	92.49	96.54	34.58	03.00



(a) Overall performance comparison based on $\mathcal{D}_\alpha$ dataset

(b) Overall performance comparison based on $\mathcal{D}_\beta$ dataset

Fig. 7: Comparison of performance with baseline algorithms

TABLE VIII: Comparison of performance with state-of-the-art methods.

Authors	Technique	Dataset	Accuracy
Singh et al. [32]	DaaS	UNSW-NB15	95.00%
Singh et al. [33]	DBM	KDD99	88.59%
Alsaedi et al. [34]	CART	$\mathcal{D}_\alpha$	77.00%
He et al. [35]	DVB	KDD99	91.95%
Wang et al. [36]	TS-PADM	Internet ads	97.45%
Poposed Work	ABiLSTM	$\mathcal{D}_\alpha$ (Original)	99.07%
		$\mathcal{D}_\alpha$ (Transformed)	99.09%
		$\mathcal{D}_\beta$ (Original)	98.73%
		$\mathcal{D}_\beta$ (Transformed)	98.49%

Terms & Abbreviations: DaaS: Dew Computing as a Service, DBM: Deep Boltzmann Machine, CART: Classification and Regression Trees, DVB: Distributed Variational Bayes, TS-PADM: Time Series Probability statistics-based Anomaly Detection Model.

D. Comparisons with state-of-the-art approaches

Table VIII compares the proposed ABiLSTM-based attack detection module to existing state-of-the-art methods in terms of accuracy. It can be seen that most current methods employed obsolete data sources such as KDD99, UNSW-NB15, and Power-dataset, which do not involve new attacks and hence have less significance in designing effective attack detection modules. Two different sources i.e., $\mathcal{D}_\beta$ and $\mathcal{D}_\alpha$ are used to assess the performance of our work. Furthermore, we can observe that, when compared to [32], [34], [33], [35], [36], the ABiLSTM-based attack detection module has the highest accuracy with both original and modified datasets. We suggest multiple viewpoints based on its potential design to reveal why the ABiLSTM-based attack detection module showed better results and outperforms other techniques for protecting DT data and identifying abnormal behaviours. First, by converting DT data into another format, the data pre-processing module can filter and transform data in an unsupervised manner, and second using the encoded data to validate threat detection using the ABiLSTM technique is an example of measuring the effectiveness of the attack detection system.

IV. Conclusion

In this article, we explored the IoV network and introduced a new communication framework called STIoV, which is powered by digital twin technology to ensure security and trustworthiness. We mapped physical RSU networks into a virtual environment to create a vehicular relationship model between IoV. Additionally, we proposed a mutual authentication and key agreement scheme to make sure IoV communication is secure and strong. To achieve transparent trust evaluation, we also suggested a new blockchain-based credit-oriented trust management system. The DT model was used by the proposed DL-based IDS to detect intrusion. Lastly, we proposed a novel blockchain-based transaction writing scheme where the cloud server writes transaction value into a blockchain to guarantee privacy and integrity. In future work, we plan to deploy proposed framework on a real-world DT-empowered IoV network, and test and improve our method on the main Ethereum network.

Acknowledgments

This research is supported by Prince Mohammad Bin Fahd Center for Futuristic Studies (PMFCFS) grant at the Prince Mohammad Bin Fahd University, Saudi Arabia.

References

[1] M. Kamal, G. Srivastava, and M. Tariq, "Blockchain-based lightweight and secured v2v communication in the internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 3997–4004, 2020.

[2] H. Chai, S. Leng, Y. Chen, and K. Zhang, "A hierarchical blockchain-enabled federated learning algorithm for knowledge sharing in internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 3975–3986, 2020.

[3] J. Wang, K. Zhu, and E. Hossain, "Green internet of vehicles (iov) in the 6g era: Toward sustainable vehicular communications and networking," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 1, pp. 391–423, 2021.

[4] S. K. Singh, R. Singh, and B. Kumbhani, "The evolution of radio access network towards open-ran: Challenges and opportunities," in *2020 IEEE wireless communications and networking conference workshops (WCNCW)*. IEEE, 2020, pp. 1–6.

[5] A. S. Abdalla, P. S. Upadhyaya, V. K. Shah, and V. Marojevic, "Toward next generation open radio access networks: What o-ran can and cannot do!" *IEEE Network*, vol. 36, no. 6, pp. 206–213, 2022.

[6] P. Kumar, G. P. Gupta, and R. Tripathi, "An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for iomt networks," *Computer Communications*, vol. 166, pp. 110 – 124, 2021.

[7] K. S. Sahoo and D. Puthal, "Sdn-assisted ddos defense framework for the internet of multimedia things," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, no. 3s, pp. 1–18, 2020.

[8] E. Bellini, Y. Iraqi, and E. Damiani, "Blockchain-based distributed trust and reputation management systems: a survey," *IEEE Access*, vol. 8, pp. 21 127–21 151, 2020.

[9] G. Fortino, L. Fotia, F. Messina, D. Rosaci, and G. M. Sarné, "Trust and reputation in the internet of things: State-of-the-art and research challenges," *IEEE Access*, vol. 8, pp. 60 117–60 125, 2020.

[10] A. Goel, A. Sharma, D. Gupta, and A. Khanna, "Immigration control and management system using blockchain," *Available at SSRN 3564189*, 2020.

[11] P. P. Ray, D. Dash, K. Salah, and N. Kumar, "Blockchain for iot-based healthcare: background, consensus, platforms, and use cases," *IEEE Systems Journal*, vol. 15, no. 1, pp. 85–94, 2020.

[12] G. S. Aujla and A. Jindal, "A decoupled blockchain approach for edge-envisioned iot-based healthcare monitoring," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 491–499, 2021.

[13] J. Xu, K. Xue, S. Li, H. Tian, J. Hong, P. Hong, and N. Yu, "Healthchain: A blockchain-based privacy preserving scheme for large-scale health data," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8770–8781, 2019.

[14] W. Wang, Q. Chen, Z. Yin, G. Srivastava, T. R. Gadekallu, F. Alsolami, and C. Su, "Blockchain and puf-based lightweight authentication protocol for wireless medical sensor networks," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8883–8891, 2021.

[15] B. Deebak, F. H. Memon, S. A. Khowaja, K. Dev, W. Wang, N. M. F. Qureshi, and C. Su, "A lightweight blockchain-based remote mutual authentication for ai-empowered iot sustainable computing systems," *IEEE Internet of Things Journal*, vol. 10, no. 8, pp. 6652–6660, 2022.

[16] B. D. Deebak, F. H. Memon, K. Dev, S. A. Khowaja, W. Wang, and N. M. F. Qureshi, "Tab-sapp: A trust-aware blockchain-based seamless authentication for massive iot-enabled industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 243–250, 2023.

[17] M. Usman, M. A. Jan, X. He, and J. Chen, "P2dca: A privacy-preserving-based data collection and analysis framework for iomt applications," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1222–1230, 2019.

[18] S. Khan and A. Akhunzada, "A hybrid dl-driven intelligent sdn-enabled malware detection framework for internet of medical things (iomt)," *Computer Communications*, vol. 170, pp. 209–216, 2021.

[19] S. P. RM, P. K. R. Maddikunta, M. Parimala, S. Koppu, T. R. Gadekallu, C. L. Chowdhary, and M. Alazab, "An effective feature engineering for dnn using hybrid pca-gwo for intrusion detection in iomt architecture," *Computer Communications*, vol. 160, pp. 139–149, 2020.

[20] Y. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: a survey," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13 789–13 804, 2021.

[21] Y. Dai, K. Zhang, S. Maharjan, and Y. Zhang, "Deep reinforcement learning for stochastic computation offloading in digital twin networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4968–4977, 2020.

[22] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on information theory*, vol. 29, no. 2, pp. 198–208, 1983.

[23] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2002, pp. 337–351.

[24] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1495–1505, 2018.

[25] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture on IE*, vol. 2, no. 1, pp. 1–18, 2015.

[26] C.-G. Huang, H.-Z. Huang, and Y.-F. Li, "A bidirectional lstm prognostics method under multiple operational conditions," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 11, pp. 8792–8802, 2019.

[27] R. Zhao, R. Yan, J. Wang, and K. Mao, "Learning to monitor machine health with convolutional bi-directional lstm networks," *Sensors*, vol. 17, no. 2, p. 273, 2017.

[28] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[29] N. Moustafa, "Ton_iot datasets," 2019, online; accessed 10-Feb-2020. [Online]. Available: http://dx.doi.org/10.21227/fesz-dm97

[30] I. Sharafaldin, "Cic-ids2017 datasets," 2017, online; accessed 15-Mar-2019. [Online]. Available: http://205.174.165.80/CICDataset/CIC-IDS-2017/Dataset/

[31] R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, and N. Kumar, "P2sf-iov: A privacy-preservation-based secured framework for internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2021.

[32] P. Singh, A. Kaur, G. S. Aujla, R. S. Batth, and S. Kanhere, "Daas: Dew computing as a service for intelligent intrusion detection in edge-of-things ecosystem," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[33] M. Singh, G. S. Aujla, A. Singh, N. Kumar, and S. Garg, "Deep-learning-based blockchain framework for secure software-defined industrial networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 606–616, 2021.

[34] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "Ton˙iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165 130–165 150, 2020.

[35] W. He, Y. Liu, H. Yao, T. Mai, N. Zhang, and F. R. Yu, "Distributed variational bayes-based in-network security for the internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6293–6304, 2021.

[36] F. Wang, R. Li, H. Wang, H. Zhu, and N. Xiong, "Ts-padm: Anomaly detection model of wireless sensors based on spatial-temporal feature points," *Wireless Communications and Mobile Computing*, vol. 2021, 2021.