An Automated Threat Intelligence Framework for Vehicle Road Cooperation Systems

Prabhat Kumar, Randhir Kumar, Alireza Jolfaei and Nazeeruddin Mohammad

***Abstract*—Vehicle Road Cooperation Systems (VRCS) use next-generation Internet technologies, including 5G, edge computing, and artificial intelligence to improve mobility, comfort, and travel efficiency. Internet of Vehicles (IoV) ecosystem serves as the technological backbone for VRCS by enabling seamless communication and data exchange between vehicles, infrastructure, and traffic management centers. This enables real-time, high-speed communication, efficient data processing, and enhanced security, fostering the development of autonomous driving, smart traffic management, and seamless connectivity within the VRCS ecosystem. At the same time, cyber attacks have become more complex, persistent, organized, and weaponized in IoV network. Threat Intelligence (TI) has emerged as a prominent security approach to obtain a complete view of the dynamically growing cyber threat environment. On the other hand, modeling TI is a challenging task due to the limited labels available for different cyber threat sources. Second, most of the available designs requires a large investment of resources and use hand-crafted features, making the entire process error-prone and time-consuming. To tackle these challenges, this paper presents TIMIF, a deep-learning-based threat intelligence modeling and identification framework for Intelligent IoV and is based on three key modules: first, the proposed TIMIF adopts an Automated Pattern Extractor (APE) module to extract hidden patterns from IoV networks. Employing its output, we design a TI-Based Detection (TIBD) module to detect abnormal behavior and TI-Attack Type Identification (TIATI) module to identify attack types. Extensive experiments are carried out on three different publicly intrusion data sources namely HCRL-car hacking, ToN-IoT and CICIDS-2017 to illustrate the utility of TIMIF framework over some commonly used baselines and state-of-the-art techniques.**

***Index Terms*—Cyber threats, Deep Learning (DL), Internet of Vehicles (IoV), Threat Intelligence (TI), Vehicle Road Cooperation Systems**

I. Introduction

Vehicle-Road Cooperation Systems (VRCS) represent an innovative approach to improving road safety and traffic management by fostering collaboration between vehicles and the surrounding road infrastructure [1]. The Internet of Vehicles (IoV) plays a pivotal role in enabling VRCS. IoV encompasses a network of interconnected vehicles, roadside sensors, and traffic management centers, facilitating real-time data exchange and communication [2]. Thus, IoV creates an intelligent integrated network to exchange information directly or indirectly between vehicles and public networks using Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication to make an efficient, safer and improved transportation system [3]. Recently, with the emergence of 5G-enable communication network, edge computing, and AI, there has been a significant growth in the number of users and it is anticipated that IoV will provide $1400 U.S. benefits per vehicle, per year [4]. However, due to the IoV openness, particularly data exchange between IoV, infrastructure, and traffic management centers, it is frequently targeted by malware (e.g., the Trojan horse and the worm) and some major cyberattacks, such as Denial of Service (DoS) and Distributed Denial of Service (DDoS), pose significant threats with profound implications for the memory and computational capabilities of vehicles. DoS attacks involve overwhelming a vehicle's communication channels or processing resources, rendering it unable to respond to legitimate requests [5]. DDoS attacks, on the other hand, amplify the impact by coordinating multiple compromised devices to flood a vehicle's systems. These attacks not only disrupt critical V2V and V2I communication in IoV ecosystem, but also strain the memory and computational resources of the vehicle's onboard systems. The implications are twofold: First, they can lead to delays or failures in essential safety-critical functions like collision avoidance, putting lives at risk [6]. Second, the increased computational load can drain the vehicle's resources, potentially leading to reduced performance and safety compromises. Securing IoV systems within the IoV ecosystem is of utmost importance to guarantee the safety and security of connected and autonomous vehicles within our ever-expanding and interconnected transportation network [7].

Recent cyberattacks have brought to the forefront the shortcomings of conventional security methodologies, including firewalls and intrusion detection and prevention systems [8]. These traditional approaches hinge on heuristic and passive attack patterns, rendering them ineffective in detecting novel threat variants [9], [10]. Consequently, organizations worldwide are exhibiting an elevated willingness to leverage the collaborative exchange of Threat Intelligence (TI). This approach enables them to cultivate a comprehensive understanding of the swiftly evolving cyber threat landscape and fortify their defenses against persistent, sophisticated, organized, and weaponized cyberattacks. TI refers to any information that can assist a company in identifying, assessing, monitoring, and responding to cyber-threats [11]. In the age of big data, it is crucial to remember that modeling TI refers to information that has been gathered, evaluated, mined, and transformed into a set of strategies or preventive measures that can be taken in

Prabhat Kumar is with the Department of Software Engineering, LUT School of Engineering Science, LUT University, 53850 Lappeenranta, Finland (Email: prabhat.kumar@lut.fi).

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

Alireza Jolfaei is with the 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).

response to a current or evolving cyber-threats ahead of time. For example, spam URLs, brute force login threats, botnet node and malware activities are potential threats to domain name infrastructure component. Identifying the attack types of a network element improves both early threat warning and specific defensive actions [12].

II. System Models

The applicable network and threat models that have been utilized to create and assess the proposed TIMIF framework are discussed briefly in this subsection.

1) Network Model: The VRCS network is shown in Fig. 1 that consist three layers: the physical layer, virtual layer, and management layer. The physical layer includes various personal devices, Road Side Units (RSUs), intelligent vehicles equipped with ambient, backscatter sensors. Each vehicle has its own On-Board Unit (OBU) and IoT devices that collect environmental data and captures various critical events such as vehicles situations, conditions of surrounding environment and driving patterns. These entities are the primary data sources for TIMIF, where the Automated Pattern Extractor (APE) module first interacts with the network. The assumption here is that despite the resource constraints of On-Board Units (OBUs) in vehicles, they can still effectively capture and communicate critical data related to environmental conditions and potential security events. This data is essential for the initial pattern extraction process, which seeks to identify hidden patterns indicative of cyber threats. At the virtual layer, edge servers, equipped with high-performance computing resources, are tasked with the initial processing of data collected from the physical layer. This layer is crucial for TIMIF, particularly for the TI-Based Detection (TIBD) module, which utilizes the processed data to detect abnormal behaviors indicative of cyber threats. The proximity of edge servers to the data sources reduces communication delays, ensuring timely threat detection. The management layer encompasses cloud centers that are responsible for more complex processing and storage tasks. This layer supports the TI-Attack Type Identification (TIATI) module by providing the computational resources needed to analyze the vast amounts of data and identify specific attack types. The assumption here is the layer's ability to maintain end-to-end elasticity, ensuring that the TIMIF framework can scale according to the volume of data and complexity of threats encountered. Throughout the network model, we assume that communication occurs over potentially insecure channels, a reality that underscores the importance of TIMIF's role in the IoV ecosystem. This assumption drives the necessity for robust threat detection and identification mechanisms capable of operating effectively despite the inherent vulnerabilities of these communication pathways.

2) Threat Model: In the domain of cyber security, particularly within the scope of TI for the IoV, the selection of a robust and comprehensive threat model is paramount. Our research employs the commonly "Dolev-Yao (DY)" threat model [13]. According to the DY model, all participating entities in the network model exchange the information over an insecure (public) channel using Internet. An attacker $\mathcal{A}$

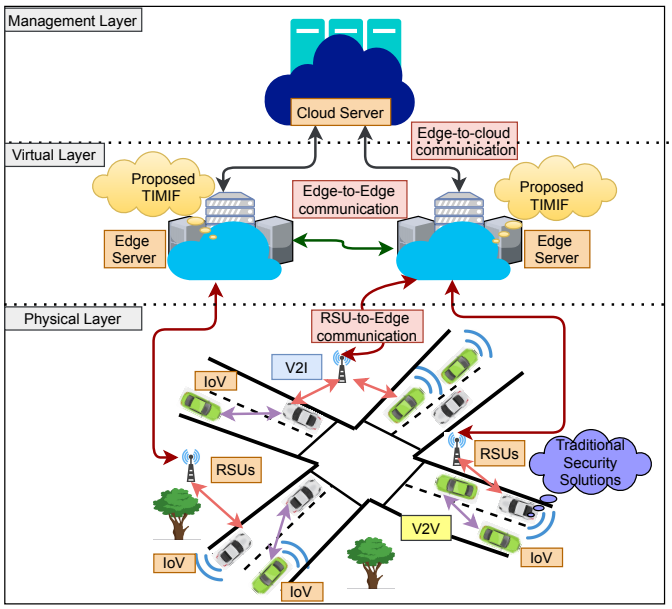


Fig. 1: Network Model of TIMIF framework

can intercept the communicated information, and also can modify or delete the internal contents of messages that are already in network. Moreover, $\mathcal{A}$ can deploy its own botnets and can launch various attacks such as ransomware, DoS, SSH-Patator, FTPPatator, DoS-Hulk, DoS-Slowhttptest, DoS-Goldeneye, scanning, injection, man-in-the-middle attack and so on in IoV network [14]. As a result, legitimate users of the IoV environment may be barred to obtain data from smart equipment, or the data may be released or modified in an illegal way. Hence, it is important to safeguard IoV communication from such threats.

A. Motivation

The most basic requirements for any cyber threat early detection and warning system should undoubtedly involve TI modelling and threat type identification for a network. Due to their effective ability to comprehend vast amounts of data and counteract hidden threat occurrences, artificial intelligence (AI)-based TI models have attracted significant attention from business and academia [15], [16]. However, there are three key limitations that prevents faster realization of AI-based TI models. First, there is limited threat type labels available for data generated from infrastructure nodes involved in TI. Manual effort or labeling is still heavily used in threat type analysis to provide insight of attack behaviors, and more significantly, to generate TI for attack detection. Moreover, manual labeling is time consuming and error-prone procedure that necessitates a large investment of resources [17]. As a result, most security analysts and companies are concerned about how to reliably and effectively learn from limited labeled infrastructure node data. Second, for developing intelligent TI models, the majority of previous research has relied on statistical, traditional Machine Learning (ML) and supervised learning techniques. However, their models were complicated, had low detection accuracy, high false alarm rates and lacked

generalization capabilities, making them challenging to use against dynamic threats [18]. Third, most of the previous security approaches based on ML and DL techniques collects and analyses network data from OBU or RSU to detect threats. However, OBU and RSU have limited resources and therefore, it impedes the network managers to sample adequate network data for threat detection and identification [7].

B. Research Contribution

In designing of the efficient and reliable TI models, few existing works adopted Deep-Learning (DL)-based approach. However, these techniques used device data (e.g., opcodes) that can cause late threat detection. On the other hand, proposed TIMIF framework uses network traffic to analyze and detect threats in IoV network. To the best of authors knowledge, we are the first to design an automated TI modeling and identification framework for IoV network. This work makes several significant contributions.

- A novel automated threat intelligence modeling and identification framework named, TIMIF is designed using DL techniques to efficiently detect threats in IoV network.
- An unsupervised and Automated Pattern Extractor (APE) module is designed using proposed Bidirectional Long Short-Term Memory Variational AutoEncoder (BLSTM-VAE) algorithm to extract hidden patterns from IoV network traffic datasets. Essentially, the proposed approach solves the vanishing gradient problem and retains information in an unidirectional manner, thereby, extending the capabilities of standard VAE.
- A TI-Based Detection (TIBD) module is designed using Bidirectional Gated Recurrent Unit (BGRU) algorithm to detect abnormal instances. A TI-Attack Type Identification (TIATI) module is designed by combining Extended Self Attention (ESA) mechanism with Deep Bidirectional Gated Recurrent Unit (ESA-DBGRU). This approach selects sequences that is more relevant to extract more discriminative features, and helps in identifying specific attack types from IoV network, and thereby improves overall detection and prediction accuracy.
- The proposed TIMIF framework is deployed at virtual layer to collect network data from edge servers instead OBU or RSU. The underlying approach is evaluated on three different data sources of the HCRL-car hacking ($\mathcal{O}_\alpha$) [19], ToN-IoT ($\mathcal{O}_\beta$) [20] and CICIDS-2017 ($\mathcal{O}_\gamma$) [21]. The experimental results are obtained by implementing both the existing state-of-the-art and some commonly used baseline techniques within our environment. This implementation allows for a direct comparison, demonstrating enhanced performance in identifying threat types.

The remainder of the paper is laid out as follows: The Section III discusses the relevant study. The proposed TIMIF framework and its components are depicted in Section IV. The suggested approach's performance analysis and comparison with the baseline and state-of-the-art methodologies are described in Section V. Finally, Section VI summarizes this paper by outlining future research.

III. Related Work

Threat modeling and identification based on ML and DL techniques in IoV network is an emerging field. We were unable to find any research work that could be directly compared to the proposed approach. On the other hand, our study draws on a variety of ongoing studies that utilise ML and DL-based TI solutions. For instance, Koloveas et al. [11] offered a machine learning (ML)-based system called "intime" that allowed security teams to identify, aggregate, evaluate, and exchange TI from several well-known social networks. Although this model employed a typical ML technique, it only managed to attain a low overall accuracy of 87.51%. To identify risks from the dark web social network, the authors of [12] employed the Multi-Layer Perceptron (MLP) model and doc2vec as a threat extraction tool. Overall accuracy for the suggested model was 79.4%. A ML-based TI framework for Industrial Control System (ICS) was developed by Atluri et al. [16] and obtained 94.24% accuracy using the Bagging Decision Trees (BDT) model. Usman et al. [17] introduced a unique method to detect malicious Internet Protocol (IP) during communication utilising dynamic malware analysis, TI, ML, and data forensics. Several ML models were used to access the effectiveness of the underlying strategy, with Decision Tree (DT) obtaining 93.5% correct predictions. Noor et al. [18] suggested an ML-based security framework to identify cyber risks based on observed attack patterns and achieved 92% accuracy. With the use of Classification and Regression Trees (CART) for IoT networks, Alsaedi et al. [14] developed a data-driven security solution and achieved 88.00% accuracy using the $\mathcal{O}_\beta$ dataset.

Various researchers designed and surveyed security architectures for IoV and considered the requirement of an efficient and secured communication at the virtual layer as a challenging issue [2], [3], [4]. Considering this various researchers have studied, and discussed the advantage of DL and statistical approaches in designing effective TI scheme. For instance, Nie et al. [7] presented a deep Convolutional Neural Network (CNN)-assisted security approach for IoV. This scheme obtained accuracy of 97.60% by utilizing the link loads of RSUs. Moustafa et al. [15] designed a TI scheme that worked on two key modules. First, a smart data management module was designed to handle heterogeneous data sources and TI technique using beta Mixture-Hidden Markov Models (MHMMs) statistical technique was designed to monitor and recognise cyber-attacks in industrial settings. The proposed scheme used Independent Component Analysis (ICA) as feature extraction tool and performance was evaluated using UNSW-NB15 and Power system dataset and model achieved 98.45% and 96.32% accuracy, respectively. Zhang et al. [5] proposed HPPELM for detecting intrusion in IoV and evaluated their framework using NSL-KDD and CIC-IDS-2017 datasets. Haddaji et al. [6] integrated Federated Learning (FL) with Transfer Learning (TL) withinn Controller Area Network (CAN) of IoV for detecting attacks using HCRL-car hacking dataset. Similarly, authors in [22] and [23] also used HCRL-car hacking dataset to proposed security mechanism using deep learning technique.

This article has been accepted for publication in IEEE Internet of Things Journal. 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/JIOT.2024.3397652

$$log\mathcal{P}_\theta(\overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}},\overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}}) = \log\int \mathcal{P}_\theta\left(\overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}},\overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}} \mid \overrightarrow{\mathcal{B}^{(1:\mathcal{T})}},\overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}}\right)\mathcal{P}\left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}},\overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}}\right)d\left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}},\overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}}\right), \tag{1}$$

$$= \log\int \mathcal{Q}_\phi\left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}},\overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}} \mid \overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}},\overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}}\right)\frac{\mathcal{P}_\theta\left(\overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}},\overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}} \mid \overrightarrow{\mathcal{B}^{(1:\mathcal{T})}},\overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}}\right)\mathcal{P}\left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}},\overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}}\right)}{\mathcal{Q}_\phi\left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}},\overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}} \mid \overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}},\overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}}\right)}d\left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}},\overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}}\right), \tag{2}$$

$$\geq \int \mathcal{Q}_\phi\left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}},\overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}} \mid \overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}},\overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}}\right)\log\frac{\mathcal{P}_\theta\left(\overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}},\overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}} \mid \overrightarrow{\mathcal{B}^{(1:\mathcal{T})}},\overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}}\right)\mathcal{P}\left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}},\overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}}\right)}{\mathcal{Q}_\phi\left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}},\overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}} \mid \overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}},\overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}}\right)}d\left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}},\overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}}\right), \tag{3}$$

$$= \int \mathcal{Q}_\phi\left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}},\overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}} \mid \overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}},\overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}}\right)\left\{\log\frac{\mathcal{P}\left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}},\overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}}\right)}{\mathcal{Q}_\phi\left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}},\overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}} \mid \overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}},\overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}}\right)}\right.$$
$$\left. + \log\mathcal{P}_\theta\left(\overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}},\overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}} \mid \overrightarrow{\mathcal{B}^{(1:\mathcal{T})}},\overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}}\right)d\left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}},\overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}}\right)\right\}, \tag{4}$$

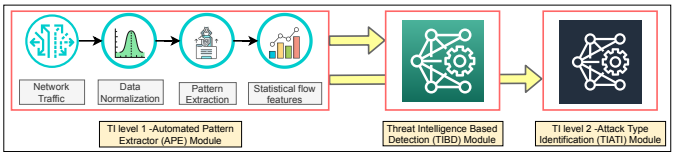


Fig. 2: High-level architecture of proposed TIMIF framework

IV. Our Proposed TIMIF Framework

A. Problem Formulation

Let us consider from the physical layer of IoV network $\mathcal{N}\times\mathcal{T}$ data records is constructed such that $\mathcal{O} = \left\{\left(\mathcal{O}_\mathcal{L}^{(1)},\mathcal{Y}^{(1)}\right),\left(\mathcal{O}_\mathcal{L}^{(2)},\mathcal{Y}^{(2)}\right),\left(\mathcal{O}_\mathcal{L}^{(\mathcal{T})},\mathcal{Y}^{(\mathcal{T})}\right)\right\}$, $\mathcal{O}_\mathcal{L}^{(\mathcal{I})}\in\mathbb{R}^\mathcal{N}$, where $\mathcal{T}$ is the number of data sample and $\mathcal{N}$ is the length of each data sample. The subscript $\mathcal{L}$ denotes that it is labeled data. Divide the the labeled training set $\mathcal{O}$ into 70% unlabeled data set $\widehat{\mathcal{O}} = \mathcal{O}_\mathcal{U}^{(1)},\mathcal{O}_\mathcal{U}^{(2)},\ldots,\mathcal{O}_\mathcal{U}^{(\mathcal{T})}\in\mathbb{R}^\mathcal{N}$ by removing the labels and construct testing set from remaining 30% labeled set as $\mathcal{O} = \mathcal{O}_\mathcal{L}^{(1)},\mathcal{O}_\mathcal{L}^{(2)},\ldots,\mathcal{O}_\mathcal{L}^{(\mathcal{T})}\in\mathbb{R}^\mathcal{N}$. The corresponding labels $\mathcal{Y}^{(\mathcal{I})}\in\{+1,-1\}$ will be used to identify abnormal behaviors and $\mathcal{Y}^{(\mathcal{I})}\in\{1,2,\ldots,\mathcal{C}\}$ will be used to identify the attack types. Then, each sample $\mathcal{O}$ is normalized using min-max normalization technique, discrete features are encoded into numeric by using label encoding technique and missing values are replaced by their column mean value. Our goal is to use the unlabeled data $\widehat{\mathcal{O}}$, and extract unknown threat patterns. Further, the extracted data is used to analyse dynamic and heterogeneous IoV traffic data to improve accuracy, detection rate and decrease false alarm rate. In order to achieve above goal, we design, and implement TIMIF framework that uses APE module to extract threat patterns automatically, and further the extracted data is feed into TIBD module, and TIATI module to detect threat and its types, respectively. A high-level architecture of TIMIF framework is illustrated in Fig 2. Each module and its working is discussed below:

B. TI level 1 -Automated Pattern Extractor (APE) Module

An Automated Pattern Extractor (APE) module based on DL technique is proposed that extracts spatial features and temporal patterns from original network data in an automated manner. This module takes discriminative spatial measurements and determines feature relationships, resulting in pattern notions that are simple and helpful. A BLSTM-VAE algorithm is proposed to design the APE module, as illustrated in Algorithm 1. The BLSTM-VAE is an unsupervised form of VAE in which BLSTM networks serve as encoders and decoders. Essentially, the proposed approach solves the vanishing gradient problem and retains information in a unidirectional manner, thereby, extending the capabilities of standard VAE. The sequential inputs of IoV data can be encoded as an equal-length sequence of latent variables using the BLSTM-VAE technique [24]. The model consists of two parts: an *encoder En* for mapping the input data $\widehat{\mathcal{O}}^{(1:\mathcal{T})}$ to a hidden latent variable $\mathcal{B}^{(1:\mathcal{T})}$ by processing samples from $\widehat{\mathcal{O}}^{(1)}$ to $\widehat{\mathcal{O}}^{(\mathcal{T})}$ in forward direction and processing samples in backward direction $\widehat{\mathcal{O}}^{(\mathcal{T})}$ to $\widehat{\mathcal{O}}^{(1)}$ and a *decoder De* that maps $\mathcal{B}^{(1:\mathcal{T})}$ back to $\widehat{\mathcal{O}}^{(1:\mathcal{T})'}$ by processing samples from $\mathcal{B}^{(1)}$ to $\mathcal{B}^{(\mathcal{T})}$ in forward direction and processing samples in backward direction $\mathcal{B}^{(\mathcal{T})}$ to $\mathcal{B}^{(1)}$, such that $\widehat{\mathcal{O}}^{(1:\mathcal{T})'}$ is approximately equal to $\widehat{\mathcal{O}}^{(1:\mathcal{T})}$. We define a Gaussian prior $\mathcal{P}_\theta\left(\mathcal{B}^{(1:\mathcal{T})}\right) = \mathcal{N}\left(\mathcal{B}^{(1:\mathcal{T})} \mid 0,\mathcal{I}\right)$ on $\mathcal{B}^{(1:\mathcal{T})}$, and can rewrite *En* as $\mathcal{Q}_\phi(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}},\overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}} \mid \overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}},\overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}})$ parameterized by ϕ and *De* as $\mathcal{P}_\theta(\overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}},\overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}} \mid \overrightarrow{\mathcal{B}^{(1:\mathcal{T})}},\overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}})$ parameterized by θ, where $\widehat{\mathcal{O}}^{(\mathcal{T})}$ denotes the last data frame. The parameter set ϕ of *En* and θ of *De* are trained by maximizing the lower bound of the log periphery likelihood $\log\mathcal{P}_\theta(\overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}},\overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}})$ and computed using Eq.(1) -Eq.(6). Then Eq.(2) is transformed into Eq.(3) by using Jensen inequality technique [25]. In Eq.(6) the first term is a logarithmic reconstruction likelihood and second term is Kullback–Leibler (KL) divergence, $\mathcal{P}(\mathcal{B}^{(1:\mathcal{T})},\mathcal{B}^{(\mathcal{T}:1)}) = \mathcal{N}((\mathcal{B}^{(1:\mathcal{T})},\mathcal{B}^{(\mathcal{T}:1)}) \mid$

This article has been accepted for publication in IEEE Internet of Things Journal. 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/JIOT.2024.3397652

$$= \int \mathcal{Q}_\phi \left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}}, \overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}} \mid \overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}}, \overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}} \right) \log \mathcal{P}_\theta \left(\overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}}, \overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}} \mid \overrightarrow{\mathcal{B}^{(1:\mathcal{T})}}, \overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}} \right) d \left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}}, \overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}} \right)$$
$$- \int \mathcal{Q}_\phi \left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}}, \overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}} \mid \overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}}, \overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}} \right) \log \frac{\mathcal{Q}_\phi \left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}}, \overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}} \mid \overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}}, \overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}} \right)}{\mathcal{P} \left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}}, \overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}} \right)}, \tag{5}$$

$$= \mathcal{B}_{\mathcal{Q}_{\phi}\left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}}, \overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}} \mid \overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}}, \overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}} \right)} \left[\mathcal{P}_\theta \left(\overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}}, \overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}} \mid \overrightarrow{\mathcal{B}^{(1:\mathcal{T})}}, \overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}} \right) \right]$$
$$- D_{KL} \left(\mathcal{Q}_\phi \left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}}, \overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}} \mid \overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}}, \overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}} \right) \parallel \mathcal{P} \left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}}, \overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}} \right) \right), \tag{6}$$

$$log \mathcal{P}_\theta \left(\overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}}, \overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}} \right) \simeq \frac{1}{\mathcal{T}} \sum_{t=1}^{\mathcal{T}} \log \mathcal{P}_\theta \left(\overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}}, \overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}} \mid \overrightarrow{\mathcal{B}^{(1:\mathcal{T})}}, \overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}} \right)$$
$$- D_{KL} \left(\mathcal{Q}_\phi \left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}}, \overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}} \mid \overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}}, \overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}} \right) \parallel \mathcal{P} \left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}}, \overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}} \right) \right), \tag{7}$$

$0, \mathcal{I})$, $\mathcal{P}_\theta(\overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}}, \overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}} \mid \overrightarrow{\mathcal{B}^{(1:\mathcal{T})}}, \overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}}) = \mathcal{N}(\overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}}, \overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}}; \mu_\theta, \sigma_\theta^2)$. The logarithmic reconstruction likelihood is calculated based on sampling approximation mentioned in Eq.(7). Assuming the number of samples is $\mathcal{T}$. The latent vector $(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}}, \overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}})$ is calculated from the mean vector $\mu_\theta(\overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}}, \overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}})$ and the variance vector $\sigma_\phi^2(\overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}}, \overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}})$ using the Eq.(1) reparametarization trick.

$$\left(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}}, \overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}} \right) = \mu_\theta \left(\overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}}, \overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}} \right)$$
$$+ \sigma_\phi \left(\overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}}, \overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}} \right) \odot \epsilon \left(\epsilon \sim \mathcal{N}\left(0, \mathcal{I}\right) \right), \tag{8}$$

The loss function of BLSTM-VAE can be written as,

$$\mathcal{L} \left(\theta, \phi; (\overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}}, \overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}}); \mathcal{C} \right) = -log \mathcal{P}_\theta \left(\overrightarrow{\widehat{\mathcal{O}}^{(1:\mathcal{T})}}, \overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T}:1)}} \right). \tag{9}$$

where $\mathcal{C}$ is class label. The *En* latent variable $(\overrightarrow{\mathcal{B}^{(1:\mathcal{T})}}, \overleftarrow{\mathcal{B}^{(\mathcal{T}:1)}})$ is used as a feature extractor on the input measurements, allowing us to decrease the dimensionality of the original data and get low-dimensional data. The BLSTM-VAE parameters are optimized iteratively based on a gradient-descent algorithm with the Adam optimizer. The reduced dimension data $\widehat{\mathcal{O}}^{(1:\mathcal{T})'}$ is feed to TIBD and TIATI module for threat and its type detection.

Algorithm 1 Training Procedure of the Proposed DL-based Automated Pattern Extractor (APE) module to extract threat patterns from IoV network

Input: Unlabeled Dataset $(\widehat{\mathcal{O}}^{(1:\mathcal{T})})$; latent variable $\mathcal{B}^{(1:\mathcal{T})}$; training epochs $\mathcal{P}$; hyperparameter a; batch size $\mathcal{B}$; class label $c_i \in (\mathcal{C})$.
Output: Encoded feature matrix $(\widehat{\mathcal{O}}^{(1:\mathcal{T})'})$
Preprocess $(\widehat{\mathcal{O}}^{(1:\mathcal{T})})$: includes label encoding for discrete features, normalization, and filling in missing values.
θ : Encoder parameters;
ϕ : Decoder parameters;
repeat
  for Number of $\mathcal{P}$ **do**
    for Number of $\mathcal{B}$ **do**
      Train BLSTM-VAE using D_{KL} divergence by processing samples from $\widehat{\mathcal{O}}^{(1)}$ to $\widehat{\mathcal{O}}^{(\mathcal{T})}$ in forward direction and processing samples in backward direction $\widehat{\mathcal{O}}^{(\mathcal{T})}$ to $\widehat{\mathcal{O}}^{(1)}$ and vice-versa.
      Calculate loss $\mathcal{L}\left(\theta, \phi; \widehat{\mathcal{O}}^{(1:\mathcal{T})}, c, a\right)$ with Eq.(2).
      Back-propagate and Update $\mathcal{L}\left(\theta, \phi; \widehat{\mathcal{O}}^{(1:\mathcal{T})}, c, a\right)$ using gradients via a gradient-descent algorithm to change *En*, *De* network.
    end for
  end for
until Convergence of Eq.(2) is reached
Get the encoded latent vector $\mathcal{B}^{(1:\mathcal{T})}$ of the BLSTM-VAE to construct the new feature matrix $(\widehat{\mathcal{O}}^{(1:\mathcal{T})'})$ of IoV network. **return** New-Encoded pattern $\widehat{\mathcal{O}}^{(1:\mathcal{T})'}$

C. Threat Intelligence Based Detection (TIBD) Module

The extracted intelligence provided by APE module is used to design a DL-based TIBD module to identify threats in IoV network. In the case of unseen data, DL techniques can give more efficient generalization capabilities than traditional ML algorithms. A Bidirectional Gated Recurrent Unit (BGRU) algorithm is proposed to design TIBD module. Algorithm 2 discuss the steps used to train and test TIBD module. The previous and next frames at time $(\mathcal{T}-1)$ and $(\mathcal{T}+1)$, respectively, determine the ultimate production at time $(\mathcal{T})$ in the BGRU arrangement. To be more specific, the forward hidden state output $(\overrightarrow{\mathcal{G}^{(1)}}, \overrightarrow{\mathcal{G}^{(2)}}, \ldots, \overrightarrow{\mathcal{G}^{(\mathcal{N})}})$ and backward hidden state output $(\overleftarrow{\mathcal{G}^{(1)}}, \overleftarrow{\mathcal{G}^{(2)}}, \ldots, \overleftarrow{\mathcal{G}^{(\mathcal{N})}})$ is concatenated to compute final BGRU output [26]. The $\rightarrow$ denotes forward process and $\leftarrow$ represents backward process. The BGRU's transition function are computed as follows [27]:

$$\overrightarrow{\mathcal{G}_\mathcal{T}} = \mathcal{F} \left(\overrightarrow{\widehat{\mathcal{O}}^{(\mathcal{T})'}}, \overrightarrow{\mathcal{G}^{(\mathcal{T}-1)}}; \overrightarrow{\Theta^{GRU}} \right)$$

$$= \begin{cases} \overrightarrow{\mathcal{U}^{(\mathcal{T})}} = \sigma\left(\overrightarrow{\mathcal{V}^{(\mathcal{U})}}\overrightarrow{\widehat{\mathcal{O}}^{(\mathcal{T})'}} + \overrightarrow{\mathcal{M}^{(\mathcal{U})}}\overrightarrow{\mathcal{G}^{(\mathcal{T}-1)}} + \overrightarrow{\mathcal{B}^{\mathcal{U}}}\right), \\ \overrightarrow{\mathcal{R}^{(\mathcal{T})}} = \sigma\left(\overrightarrow{\mathcal{V}^{(\mathcal{R})}}\overrightarrow{\widehat{\mathcal{O}}^{(\mathcal{T})'}} + \overrightarrow{\mathcal{M}^{(\mathcal{R})}}\overrightarrow{\mathcal{G}^{(\mathcal{T}-1)}} + \overrightarrow{\mathcal{B}^{\mathcal{R}}}\right), \\ \overrightarrow{\mathcal{C}^{(\mathcal{T})}} = \tanh\left[\overrightarrow{\mathcal{V}^{(\mathcal{C})}}\overrightarrow{\widehat{\mathcal{O}}^{(\mathcal{T})'}} + \overrightarrow{\mathcal{M}^{(\mathcal{C})}}(\overrightarrow{\mathcal{R}^{(\mathcal{T})}} \odot \overrightarrow{\mathcal{G}^{(\mathcal{T}-1)}}) + \overrightarrow{\mathcal{B}^{\mathcal{C}}}\right], \\ \overrightarrow{\mathcal{G}^{(\mathcal{T})}} = \left(\overrightarrow{1-\mathcal{U}^{(\mathcal{T})}}\right) \odot \overrightarrow{\mathcal{G}^{(\mathcal{T}-1)}} + \overrightarrow{\mathcal{U}^{(\mathcal{T})}} \odot \overrightarrow{\mathcal{C}^{(\mathcal{T})}}. \end{cases} \tag{10}$$

$$\overleftarrow{\mathcal{G}_{\mathcal{T}}} = \mathcal{F}\left(\overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T})'}}, \overleftarrow{\mathcal{G}^{(\mathcal{T}+1)}}; \overleftarrow{\Theta^{GRU}}\right)$$

$$= \begin{cases} \overleftarrow{\mathcal{U}^{(\mathcal{T})}} = \sigma\left(\overleftarrow{\mathcal{V}^{(\mathcal{U})}}\overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T})'}} + \overleftarrow{\mathcal{M}^{(\mathcal{U})}}\overleftarrow{\mathcal{G}^{(\mathcal{T}+1)}} + \overleftarrow{\mathcal{B}^{\mathcal{U}}}\right), \\ \overleftarrow{\mathcal{R}^{(\mathcal{T})}} = \sigma\left(\overleftarrow{\mathcal{V}^{(\mathcal{R})}}\overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T})'}} + \overleftarrow{\mathcal{M}^{(\mathcal{R})}}\overleftarrow{\mathcal{G}^{(\mathcal{T}+1)}} + \overleftarrow{\mathcal{B}^{\mathcal{R}}}\right), \\ \overleftarrow{\mathcal{C}^{(\mathcal{T})}} = \tanh\left[\overleftarrow{\mathcal{V}^{(\mathcal{C})}}\overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T})'}} + \overleftarrow{\mathcal{M}^{(\mathcal{C})}}(\overleftarrow{\mathcal{R}^{(\mathcal{T})}} \odot \overleftarrow{\mathcal{G}^{(\mathcal{T}+1)}}) + \overleftarrow{\mathcal{B}^{\mathcal{C}}}\right], \\ \overleftarrow{\mathcal{G}^{(\mathcal{T})}} = \left(\overleftarrow{1+\mathcal{U}^{(\mathcal{T})}}\right) \odot \overleftarrow{\mathcal{G}^{(\mathcal{T}+1)}} + \overleftarrow{\mathcal{U}^{(\mathcal{T})}} \odot \overleftarrow{\mathcal{C}^{(\mathcal{T})}}. \end{cases} \tag{11}$$

where $\overrightarrow{\mathcal{R}^{(\mathcal{T})}}$ and $\overleftarrow{\mathcal{R}^{(\mathcal{T})}}$ are reset gate, $\overrightarrow{\mathcal{U}^{(\mathcal{T})}}$ and $\overleftarrow{\mathcal{U}^{(\mathcal{T})}}$ are update gate, $\overrightarrow{\mathcal{C}^{(\mathcal{T})}}$ and $\overleftarrow{\mathcal{C}^{(\mathcal{T})}}$ denotes candidate cell, and $\overrightarrow{\mathcal{G}^{(\mathcal{T})}}$ and $\overleftarrow{\mathcal{G}^{(\mathcal{T})}}$ are final state of forward and backward process, respectively. $\overrightarrow{\Theta^{GRU}}$ and $\overleftarrow{\Theta^{GRU}}$ denotes parameters set and are shared and learnt during model training. $\overrightarrow{\mathcal{V}^{(\mathcal{R})}}, \overrightarrow{\mathcal{V}^{(\mathcal{U})}}, \overrightarrow{\mathcal{V}^{(\mathcal{C})}}$ and $\overleftarrow{\mathcal{V}^{(\mathcal{R})}}, \overleftarrow{\mathcal{V}^{(\mathcal{U})}}, \overleftarrow{\mathcal{V}^{(\mathcal{C})}}$ denotes the weight matrices from input $\overrightarrow{\widehat{\mathcal{O}}^{(\mathcal{T})'}}$ and $\overleftarrow{\widehat{\mathcal{O}}^{(\mathcal{T})'}}$ to hidden layer for forward pass. $\overrightarrow{\mathcal{M}^{(\mathcal{R})}}, \overrightarrow{\mathcal{M}^{(\mathcal{U})}}, \overrightarrow{\mathcal{M}^{(\mathcal{C})}}$, and $\overleftarrow{\mathcal{M}^{(\mathcal{R})}}, \overleftarrow{\mathcal{M}^{(\mathcal{U})}}, \overleftarrow{\mathcal{M}^{(\mathcal{C})}}$ denotes the weight matrices between two consecutive hidden state for backward pass. $\overrightarrow{\mathcal{B}^{\mathcal{R}}}, \overrightarrow{\mathcal{B}^{\mathcal{U}}}, \overrightarrow{\mathcal{B}^{\mathcal{C}}}$ and $\overleftarrow{\mathcal{B}^{\mathcal{R}}}, \overleftarrow{\mathcal{B}^{\mathcal{U}}}, \overleftarrow{\mathcal{B}^{\mathcal{C}}}$ denotes bias term for all gates present in forward and backward process. σ and $\tanh$ denotes non-linear activation function. $\odot$ represents elementwise product between two elements. In a summary, BGRU hidden element representation $\overrightarrow{\mathcal{G}_{\mathcal{T}}}$ is the concatenated vector of forward and backward method outputs as follows:

$$\overrightarrow{\mathcal{G}_{\mathcal{T}}} = \overrightarrow{\mathcal{G}^{(\mathcal{T}-1)}} \oplus \overleftarrow{\mathcal{G}^{(\mathcal{T}-1)}} \tag{12}$$

where $\oplus$ is the elementwise summation of these corresponding elements of two vectors. The BGRU iterates the following steps (Eq. 10 to 12) in different timesteps depending on how many outputs the APE module produces during the training stage. The sigmoid function is used to move the output of BGRU layers to the output layer. The output layer consist of an *Adam* optimizer with a binary crossentropy loss function to perform the binary intrusion detection operation. The $\mathcal{LOSS}$ function is computed using Eq.13,

$$\mathcal{LOSS}\left(\mathcal{Y}^{(\mathcal{I})}, \widehat{\mathcal{Y}^{(\mathcal{I})}}\right) = -\frac{1}{\mathcal{N}}\sum_{\mathcal{I}=1}^{\mathcal{N}} \mathcal{Y}^{(\mathcal{I})} \cdot \log \widehat{\mathcal{Y}^{(\mathcal{I})}} + \left(1-\mathcal{Y}^{(\mathcal{I})}\right) \cdot \left(1-\widehat{\mathcal{Y}^{(\mathcal{I})}}\right) \tag{13}$$

In order to detect complex threat behaviors, the BGRU-based TIBD module acquires the hidden patterns retrieved from the APE module as indicated in Algorithm 2 from the IoV network at each timestep $(\mathcal{T})$. The patterns of Ransomware assaults collected from the APE module, for example, [0.47584813, 0.21475718, 0.69874172, 0.96317582, 0.78512380, 0.63637120, 0.98736715, 0.66227415, 0.74187214, 0.69742147], are sent to the TIBD module for detection of abnormal instances.

Algorithm 2 Training Procedure of the Proposed DL-based TIBD module for threat detection in IoV network

Input: APE Output as Training set: $\widehat{\mathcal{O}}^{(1:\mathcal{T})'}$; Test set $\mathcal{O}$; training epochs $\mathcal{P}$; hyperparameter a; batch size $\mathcal{B}$; class label $c_i \in (\mathcal{C})$.
Initialization $h_0(\mathcal{T}) = \widehat{\mathcal{O}}^{(1:\mathcal{T})'}, \forall : \mathcal{T} \in [1, t_n]$
Output: $Normal \rightarrow 0$, $Abnomal \rightarrow 1$,
Preprocess $\mathcal{O}$: includes label encoding for discrete features, normalization, and filling in missing values.
repeat
  for Number of $\mathcal{P}$ **do**
    for Number of $\mathcal{B}$ **do**
      for $\mathcal{T} = 1 \rightarrow t_n$ **do**
        GRU calculates $(\overrightarrow{\mathcal{G}^{(1)}}, \overrightarrow{\mathcal{G}^{(2)}}, \ldots, \overrightarrow{\mathcal{G}^{(\mathcal{N})}})$ through forward pass using Eq. (3)
        For the forward pass, use Eq. (3) to determine the reset, candidate cell, update, and final states for each time step and store output activations.
      end for
      for $\mathcal{T} = t_n \rightarrow 1$ **do**
        GRU calculates $(\overrightarrow{\mathcal{G}^{(\mathcal{N})}}, \ldots, \overrightarrow{\mathcal{G}^{(2)}}, \overrightarrow{\mathcal{G}^{(1)}})$ through backward pass using Eq. (4)
        For the backward pass, use Eq. (4) to determine the reset, candidate cell, update, and final states for each time step and store output activations
      end for
      The final BGRU output is then calculated as the concatenated vector of outputs of forward and backward processes using Eq. (5)
      $BGRU_{Train}$ model is obtained
      Compute $\mathcal{LOSS}$ according to Eq. (6).
      Use *Adam* as convergence optimizer.
    end for
  end for
until Convergence of Eq. (6) is reached
while *True* **do**
  Predict = $BGRU_{Train}$ $(\mathcal{O})$
  if *Predict.value* $==$ 0 **then**
    return Normal
  else
    return Abnormal
  end if
end while

D. TI level 2 -Attack Type Identification (TIATI) Module

The proposed TIBD module can assist security analyst in identifying abnormal behaviors, based on the patterns extracted by the APE module. However, it doesn't recognize the exact threat types of abnormal traffic. Therefore, we design a DL-based TI level 2 -TIATI module that adds an additional context to the APE module and identifies to which threat a pattern belongs. However, all patterns extracted from APE module may not contribute equally in identification of threat types. Therefore, greater attention should be assigned to more useful extracted patterns. On the other hand, the BGRU performs well in terms of sequence feature extraction, forward and backward iterations, but it neglects to pay attention to crucial sequence information, which is critical since not all segments in sequence data are equal in relevance. Identifying

the relevant sequence elements to focus on during network training is advantageous. Thus, a TIATI module is designed using Extended Self-Attention-based-Deep Bidirectional Gated Recurrent Unit (ESA-DBGRU) algorithm. Algorithm 3 discuss the steps used in designing TIATI module. In this study, the additional discriminative temporal information is extracted by employing ESA mechanism. The ESA is logical expansion of additive attention at the multi-dimensional feature level and is used to assign weights to each IoV data sample by examining the inherent significance of each sample [28]. Unlike traditional self-attention mechanisms, which assign value to each recurrent encoded slice and collate this information and construct a concluding interpretation, the ESA can effectively define the precise value by measuring the parity in each sample from separate points, and the resulting $\mathcal{K}^{(\mathcal{I})'}$ is regarded as a featurewise score vector from the *i*-th sample $\mathcal{G}^{(\mathcal{I})'}$. Additionally, in and out of the activation function two bias terms are added by the ESA mechanism. We can express the *i*-th featurewise score vector $\mathcal{K}^{(\mathcal{I})'}$ as,

$$\begin{aligned}\mathcal{K}^{(\mathcal{I})'} &= \mathcal{F}\left(\mathcal{G}^{(\mathcal{I})'}\mathcal{V}^{(\mathcal{I})}\right)\\ &= \mathcal{M}^{(\mathcal{T})}\sigma\left(\mathcal{M}^{(1)}\mathcal{G}^{(\mathcal{I})'}\mathcal{V}^{(\mathcal{I})} + \mathcal{M}^{(2)}\mathcal{V}^{(\mathcal{I})} + \mathcal{B}^{(1)}\right) + \mathcal{B},\end{aligned} \tag{14}$$

where $\mathcal{F}\left(\mathcal{G}^{(\mathcal{I})'}\mathcal{V}^{(\mathcal{I})}\right)$ denotes the intrinsic parity of the *i*-th encoded IoV data sample and based on the linear transformation of feature vector $\mathcal{G}^{(\mathcal{I})'}$, $\mathcal{V}^{(\mathcal{I})}$ represents the generated aligned pattern vector having same dimension as the input feature vector. $\sigma(.)$ denotes (*ELU*) activation function, where $\mathcal{M}$ represents the weight and $\mathcal{B}$ denotes bias terms of *ELU* activation function. The $\mathcal{M}^{(1)}$ and $\mathcal{M}^{(2)}$ denotes weights and $\mathcal{B}^{(1)}$ is the bias term. We can represent probabilities of entire samples as $\mathcal{PS} = \left\{\mathcal{PS}^{(1)}, \mathcal{PS}^{(2)}, \ldots, \mathcal{PS}^{(\mathcal{N})}\right\}$ and the probability of *i*-th IoV sample is written as:

$$\mathcal{PS}^{(\mathcal{I})} = \frac{\exp\left(\mathcal{K}^{(\mathcal{I})'(\mathcal{T})} \cdot \mathcal{G}^{(\mathcal{I})'}\right)}{\sum^{\mathcal{I}=1\to\mathcal{N}} \exp\left(\mathcal{K}^{(\mathcal{I})'(\mathcal{T})} \cdot \mathcal{G}^{(\mathcal{I})'}\right)} \tag{15}$$

Finally, the extracted features by the ESA mechanism are represented by $\mathcal{A} = \left\{\mathcal{A}^1, \mathcal{A}^2, \ldots, \mathcal{A}^{\mathcal{N}}\right\}$, and the *i*-th attentive feature retrieved by the ESA mechanism is calculated as:

$$\mathcal{A}^{(\mathcal{I})} = \mathcal{PS}^{(\mathcal{I})} \cdot \mathcal{G}^{(\mathcal{I})'}. \tag{16}$$

The softmax layer is used in the final portion of the proposed TIATI module to estimate the probability that a given pattern belongs to a certain threat category. Therefore, the retrieved spatiotemporal attentive characteristics, $\mathcal{A} = \left\{\mathcal{A}^1, \mathcal{A}^2, \ldots, \mathcal{A}^{\mathcal{N}}\right\}$, is used by the softmax function as input to identify specific threat type as:

$$\mathcal{P} = softmax\left(\mathcal{MA} + \mathcal{B}\right) \tag{17}$$

where $\mathcal{P} = \left\{\mathcal{P}^1, \mathcal{P}^2, \ldots, \mathcal{P}^{\mathcal{N}}\right\}$ is the anticipated probability of the *i*-th IoV data sample, and $\mathcal{M}$ is the weight and $\mathcal{B}$ denotes the bias components of softmax function. Finally, we calculate the cross-entropy loss to measure the output layer error across all labeled samples as:

$$\mathcal{LOSS} = -\sum_{\mathcal{I}=1}^{\mathcal{N}} \widehat{\mathcal{Y}^{(\mathcal{I})}} \log\left(\mathcal{P}^{(\mathcal{I})}\right) \tag{18}$$

where the label of the *i*-th IoV data sample is $\widehat{\mathcal{Y}^{(\mathcal{I})}}$, and the smaller cross-entropy $\mathcal{LOSS}$ implies increased threat identification accuracy. In summary, we developed a TIATI module that uses the DBGRU algorithm to investigate the temporal information of various IoV data samples, and further we incorporate the ESA mechanism to assign weight to IoV samples depending on their relevance. Finally, for IoV threat type identification, spatiotemporal attentive characteristics is acquired, which can assist a security team in performing a suitable defensive and mitigation strategy.

Algorithm 3 Training Procedure of the Proposed DL-based TIATI module for threat type identification in IoV network

Input: APE Output as Training set: $\widehat{\mathcal{O}^{(1:\mathcal{T})'}}$; Test set $\mathcal{O}$; training epochs $\mathcal{P}$; hyperparameter a; batch size $\mathcal{B}$; class label $c_i \in (\mathcal{C})$.
Initialization $h_0(\mathcal{T}) = \widehat{\mathcal{O}^{(1:\mathcal{T})'}}$, $\forall : \mathcal{T} \in [1, t_n]$
Output: $Normal \to 0$, $Attack_1 \to 1$, $Attack_2 \to 2$, $Attack_3 \to 3$, $\ldots$ $Attack_N \to N$.
Preprocess $\mathcal{O}$: includes label encoding for discrete features, normalization, and filling in missing values.
repeat
  for Number of $\mathcal{P}$ **do**
    for Number of $\mathcal{B}$ **do**
      for $\mathcal{T} = 1 \to t_n$ **do**
        GRU calculates $(\overrightarrow{\mathcal{G}^{(1)}}, \overrightarrow{\mathcal{G}^{(2)}}, \ldots, \overrightarrow{\mathcal{G}^{(\mathcal{N})}})$ through forward pass using Eq. (3)
        For the forward pass, use Eq. (3) to determine the reset, candidate cell, update, and final states for each time step and store output activations.
      end for
      for $\mathcal{T} = t_n \to 1$ **do**
        GRU calculates $(\overrightarrow{\mathcal{G}^{(\mathcal{N})}}, \ldots, \overrightarrow{\mathcal{G}^{(2)}}, \overrightarrow{\mathcal{G}^{(1)}})$ through backward pass using Eq. (4)
        For the backward pass, use Eq. (4) to determine the reset, candidate cell, update, and final states for each time step and store output activations
      end for
      The final BGRU output is obtained by concatenating outputs of forward and backward processes using Eq. (5).
      DBGRU$_{Train}$ model is obtained
      Add ESA layer to *DBGRU*$_{Train}$ using Eq. (7)-Eq. (9) to build *ESA-DBGRU* model.
      Compute $\mathcal{LOSS}$ according to Eq. (11).
      Use *Adam* as convergence optimizer
    end for
  end for
until Convergence of Eq. (11) is reached
while *True* **do**
  Predict_Attack-Type = ESA-DBGRU$_{Train}$ $(\mathcal{O})$
  Decision = Predict_Attack-Type
  return *Decision*
end while

V. Performance Analysis

This section explains the data used in the current study, including all experimental settings and procedures. We also

provide an analysis and discussion of the findings.

A. Experimental Setup

We implement TIMIF with Keras 2.3.1 using Python 3.6 programming language. During training, the batch size used was 50 with *Adam* optimizer and 10 epochs. The training procedure used Kullback–Leibler divergence, binary cross-entropy and cross-entropy loss functions for APE, TIBD and TIATI modules, respectively. The Exponential Linear Unit (ELU) activation function for hidden layer, and sigmoid for output layer of APE, TIBD and softmax for TIATI were used. The model uses 0.2 dropout rate. The findings are based on HCRL-car hacking ($\mathcal{O}_\alpha$) [19], ToN-IoT ($\mathcal{O}_\beta$) [20] and CICIDS-2017 ($\mathcal{O}_\gamma$) [21] network datasets, which comprise 701832 normal and 116608 attack instances with 5 classes, 300000 legitimate, and 161043 illegitimate occurrences having 10 classes and 2035505 legitimate and 390222 illegitimate occurrences having 11 different outcomes, respectively. The HCRL-Car Hacking Dataset ($\mathcal{O}_\alpha$) [19] is specifically generated to simulate real-world car hacking scenarios, encompassing a wide range of attacks relevant to the Internet of Vehicles (IoV) domain, such as DoS, fuzzy, gear spoofing, and RPM gauge malfunctions. Its inclusion in our experiments is pivotal due to the TIMIF framework's focus on IoV security, offering a unique opportunity to test our framework's ability to detect and classify sophisticated vehicle-specific cyber threats. The ToN-IoT dataset ($\mathcal{O}_\beta$) [20] is a comprehensive collection of IoT and IoV network traffic, including a variety of attack vectors such as DDoS, botnet, and data infiltration, among others. It represents a diverse IoT ecosystem, making it invaluable for testing the TIMIF framework's versatility in identifying and mitigating threats across different IoT and IoV scenarios. The CICIDS-2017 ($\mathcal{O}_\gamma$) [21] is the most comprehensive and recent datasets for network intrusion detection, the CICIDS-2017 dataset features a wide array of simulated real-world cyber attacks (e.g., DoS, port scans, botnets, and web attacks). The dataset's varied attack scenarios and realistic network traffic patterns enable a thorough assessment of TIMIF's detection and classification capabilities, ensuring its relevance for current and future IoV security challenges.

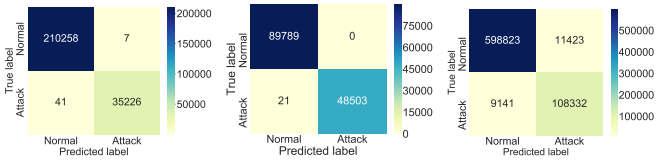
All three datasets were divided into training and testing sets, with 80% allocated for training and 20% for testing. The preprocessing steps outlined in [29] are applied. To evaluate the effectiveness of the proposed scheme, we employ a range of metrics including ACcuracy (AC), Detection Rate (DR), False Alarm Rate (FAR), PRecision (PR), and the F1 score, as specified in [29]. In order to showcase the superiority of the proposed TIMIF framework, we compare it against conventional baseline methods and existing state-of-the-art approaches.

B. Evaluation of APE-TIBD Module

The effectiveness of the proposed APE-TIBD module is evaluated based on the $\mathcal{O}_\alpha$, $\mathcal{O}_\beta$ and $\mathcal{O}_\gamma$ datasets with respect to obtained AC, DR, PR, F1, FAR, Loss, confusion matrix and ROC Curve. Table I shows the results obtained from APE-TIBD module based on various TI performance metrics. A

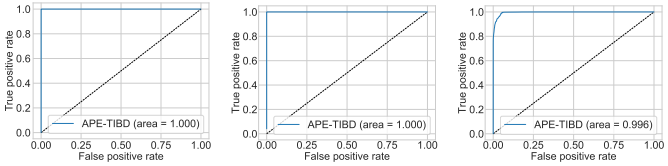
TABLE I: Result obtained from APE-TIBD module based on various performance metrics

Dataset	AC	DR	PR	F1	FAR	Loss
$\mathcal{O}_\alpha$	99.98	99.88	99.98	99.93	0.33	0.340
$\mathcal{O}_\beta$	99.98	99.95	100.00	99.97	0.00	2.09e-04
$\mathcal{O}_\gamma$	98.65	92.21	90.46	91.33	1.87	0.031



(a) The confusion matrix for $\mathcal{O}_\alpha$ dataset (b) The confusion matrix for $\mathcal{O}_\beta$ dataset (c) The confusion matrix for $\mathcal{O}_\gamma$ dataset

Fig. 3: Confusion matrices obtained from APE-TIBD module



(a) The AU-ROC Curve obtained for $\mathcal{O}_\alpha$ dataset (b) The AU-ROC Curve obtained for $\mathcal{O}_\beta$ dataset (c) The AU-ROC Curve obtained for $\mathcal{O}_\gamma$ dataset

Fig. 4: AU-ROC Curves obtained from APE-TIBD module

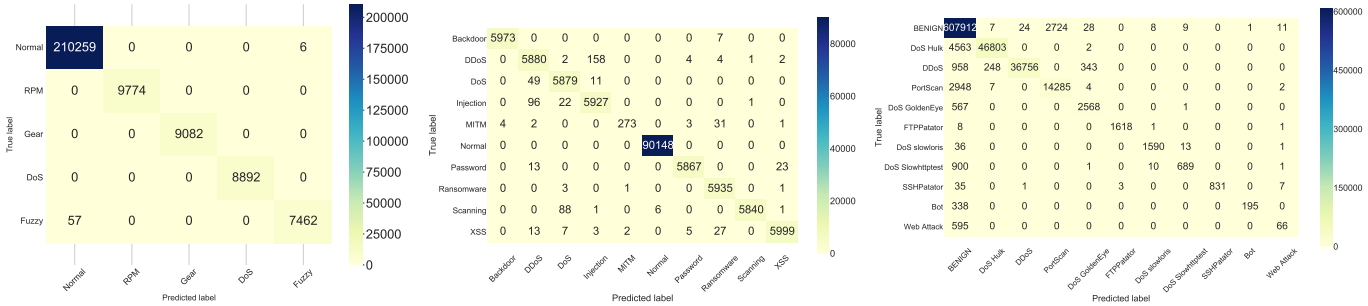
TI scheme with high accomplished values of AC, DR, PR, F1 and low FAR and loss is considered as a proficient threat detection mechanism. It can be observed, that our proposed APE-TIBD module has obtained 99.98%, 99.98% and 98.65% AC, 99.88%, 99.95% and 92.21% DR, 99.98%, 100.00% and 90.46% PR, 99.93%, 99.97% and 91.33% F1 score and has reduced FAR to 0.33%, 0.00% and 1.87% with 0.340%, 2.09e-04% and 0.031% loss using $\mathcal{O}_\alpha$, $\mathcal{O}_\beta$ and $\mathcal{O}_\gamma$ datasets, respectively.

Fig. 3 illustrates the confusion matrix obtained for APE-TIBD module to evaluate the performance. Based on the off-diagonal elements of the matrix, we see that the proposed approach has lowered the misclassification to 48, 21 and 20564 (i.e., adding False Positives (FPs) and False Negatives (FNs)) for $\mathcal{O}_\alpha$, $\mathcal{O}_\beta$ and $\mathcal{O}_\gamma$ datasets, respectively. Overall, the confusion matrix clearly depicts the excellent capability of APE-TIBD module. The area under the curve is created by plotting the True Positive (TP) and FP rates in two dimensions. The overall performance of the system is measured by the area under the curve (AU-ROC), with larger area under the curve indicating better model performance. As shown in Fig. 4, the AU-ROC value is 1.0, 1.0 and 0.996 for $\mathcal{O}_\alpha$, $\mathcal{O}_\beta$ and $\mathcal{O}_\gamma$ datasets, respectively. Thus, the obtained AU-ROC clearly indicates the out performance of the proposed algorithm. The RoC curve specifies the degree of difference between normal and attack data, making it the most significant model performance metrics.

C. Evaluation of APE-TIATI Module

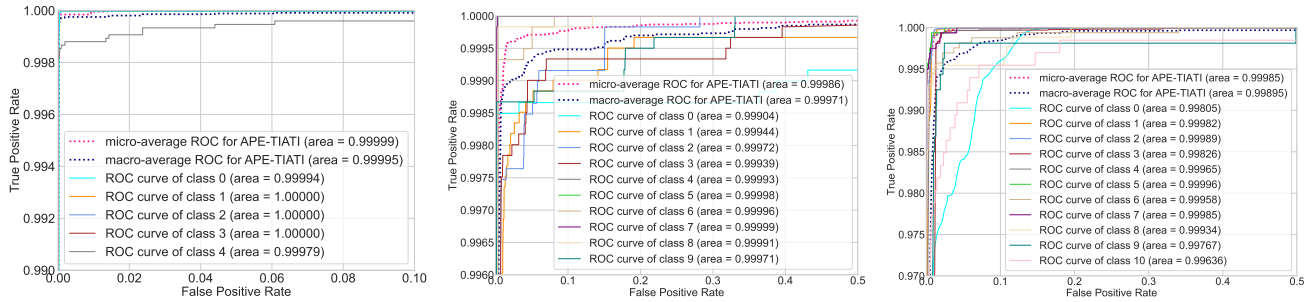
The performance of APE-TIATI module is accessed in terms of macro-averaged AC, DR, PR, F1 and per-class PR, DR,

This article has been accepted for publication in IEEE Internet of Things Journal. 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/JIOT.2024.3397652



(a) The confusion matrix obtained for $\mathcal{O}_\alpha$ dataset (b) The confusion matrix obtained for $\mathcal{O}_\beta$ dataset (c) The confusion matrix obtained for $\mathcal{O}_\gamma$ dataset

Fig. 5: Confusion matrices obtained from APE-TIATI module



(a) The AU-ROC Curve obtained for $\mathcal{O}_\alpha$ dataset (b) The AU-ROC Curve obtained for $\mathcal{O}_\beta$ dataset (c) The AU-ROC Curve obtained for $\mathcal{O}_\gamma$ dataset

Fig. 6: The AU-ROC Curve obtained from APE-TIATI module

TABLE II: Result obtained from APE-TIATI module based on various performance metrics

Dataset	AC	DR	PR	F1	Loss
$\mathcal{O}_\alpha$	99.97	99.98	99.88	99.96	0.002
$\mathcal{O}_\beta$	99.20	97.77	99.75	98.30	0.026
$\mathcal{O}_\gamma$	99.02	84.73	94.34	82.63	0.022

TABLE III: Per-class prediction results (%) for APE-TIATI module on $\mathcal{O}_\alpha$ dataset.

Parameters	Normal	RPM	Gear	DoS	Fuzzy
PR	99.97	100.00	100.00	100.00	99.91
DR	99.99	100.00	100.00	100.00	99.24
F1	99.98	100.00	100.00	100.00	99.57
FAR	0.00161	0.00	0.00	0.00	0.00002

Terms & Abbreviations: PR:Precision Rate; DR: Detection Rate; FAR: False Alarm Rate.

F1, FAR for all classes present in both datasets. Then we use validation loss, multi-class confusion matrix and AU-ROC curve. Table II shows macro-averaged results obtained from APE-TIATI module under multi-class evaluation. The numerical values for AC, DR, PR, F1 is between 97%-99% with validation loss of 0.026% using $\mathcal{O}_\alpha$, $\mathcal{O}_\beta$ and for $\mathcal{O}_\gamma$ it is between 82%-99% with 0.022% validation loss. Thus, Table II clearly depicts that the A-DBGRU based APE-TIATI module has learned benign and threat traffic signatures outstandingly. Additionally, Table III, Table IV and Table V describes the performance of APE-TIATI module in terms of per-class PR, DR, F1 and FAR. For the $\mathcal{O}_\alpha$ dataset, can identify normal and different attack types (RPM, GEar, DoS and fuzzy). For the $\mathcal{O}_\beta$ dataset, the APE-TIATI module can identify the threat types of IoV traffic belonging to recent attacks such as backdoor, DDoS, DoS, injection MITM and so on by achieving averages of 97.09%-99.93%. For the $\mathcal{O}_\gamma$ network dataset, the APE-TIATI module largely neglected to learn DoS Slowhttptes, bot and web attack. The reason behind is that the dataset has very less observations of these threats. Moreover, for rest of the classes i.e., DoS GoldenEye, FTPPatator, DoS Hulk, PortScan and DDoS attacks the proposed algorithm performed well and achieved average of 81.88%-100.00%. In Fig. 5, we have shown the confusion matrices obtained from APE-TIATI module based on $\mathcal{O}_\alpha$, $\mathcal{O}_\beta$ and $\mathcal{O}_\gamma$ datasets, respectively. Most of the occurrences in all datasets are accurately classified. Fig. 6 illustrates AU-ROC curve for each threat type obtained by APE-TIATI module using $\mathcal{O}_\alpha$, $\mathcal{O}_\beta$ and $\mathcal{O}_\gamma$ datasets, respectively. In this study, AU-ROC value is calculated and plotted for each threat and we see that for all threats present in both datasets the coloured reference line is on the top-left corner. This refers to the best performance of the model with almost 100% values for all attacks.

D. Comparison with baseline approaches

In this set of experiments, Naive Bayes (NB), Decision Tree (DT) and Random Forest (RF) techniques serves as the baseline for evaluating both TIMIF module. We may draw a number of conclusions based on the data in Table VI. First, APE-TIBD and APE-TIATI module is able to achieve higher AC (i.e., 99.98%, 99.97%, 99.98%, 99.20% and 98.65% and 99.02%) compared to NB, DT and RF using $\mathcal{O}_\alpha$, $\mathcal{O}_\beta$ and $\mathcal{O}_\gamma$ datasets, respectively. Moreover, we can observe similar trend

TABLE IV: Per-class prediction results (%) for APE-TIATI module on $\mathcal{O}_\beta$ dataset.

Parameters	Backdoor	DDoS	DoS	Injection	MITM	Normal	Password	Ransomware	Scanning	XSS
PR	99.93	98.54	97.96	97.16	97.91	99.99	98.79	98.81	99.93	99.53
DR	99.97	97.17	98.98	98.03	85.12	100.00	99.39	99.91	99.38	99.90
F1	99.64	97.15	98.66	97.22	92.54	99.15	99.59	99.44	99.16	99.61
FAR	0.00003	0.00130	0.00092	0.00130	0.00002	0.00012	0.00009	0.00052	0.00001	0.00002

Terms & Abbreviations: PR:Precision Rate; DR: Detection Rate; FAR: False Alarm Rate.

TABLE V: Per-class prediction results (%) for APE-TIATI module on $\mathcal{O}_\gamma$ dataset

Parameters	BENIGN	DoS Hulk	DDoS	PortScan	DoS GoldenEye	FTPPatator	DoS slowloris	DoS Slowhttptes	SSHPatator	Bot	Web Attack
PR	98.23	99.44	99.93	83.98	87.16	99.81	98.81	96.76	100.00	99.48	74.15
DR	99.53	91.13	95.95	82.83	81.88	99.38	96.95	43.30	94.75	36.58	09.98
F1	98.88	95.09	97.90	83.40	84.44	99.59	97.87	59.57	97.30	53.49	17.60
FAR	0.09357	0.00038	0.00003	0.00383	0.00052	0.00004	0.00002	0.00003	0.00000	0.00001	0.00003

Terms & Abbreviations: PR:Precision Rate; DR: Detection Rate; FAR: False Alarm Rate.

TABLE VI: Performance comparison with baseline algorithms

Techniques	AC	DR	PR	F1
Dataset $\mathcal{O}_\alpha$				
NB [14]	94.69	88.10	86.68	81.37
DT [17]	92.20	85.90	88.63	80.12
RF [18]	95.21	90.11	89.21	88.12
APE-TIBD	99.98	99.88	99.98	99.93
APE-TIATI	99.97	99.98	99.88	99.96
Dataset $\mathcal{O}_\beta$				
NB [14]	90.69	77.70	77.68	72.43
DT [17]	95.34	80.00	74.42	76.33
RF [18]	97.81	85.43	87.55	86.41
APE-TIBD	99.98	99.95	100.00	99.97
APE-TIATI	99.20	97.77	99.75	98.30
Dataset $\mathcal{O}_\gamma$				
NB [14]	59.57	80.65	45.75	47.00
DT [17]	98.34	62.16	77.40	64.58
RF [18]	98.62	51.77	62.01	55.62
APE-TIBD	98.65	92.21	90.46	91.33
APE-TIATI	99.02	84.73	94.34	82.63

Terms & Abbreviations: AC: Accuracy; DR: Detection Rate; PR:Precision Rate; FAR: False Alarm Rate; NB: Naive Bayes; DT: Decision Tree; RF: Random Forest; APE-TIBD: Automated Pattern Extractor-Threat Intelligence-Based Detection; APE-TIATI: Automated Pattern Extractor-Threat Intelligence Attack Type Identification.

TABLE VII: Comparison of accuracy with various other threat intelligence techniques

Authors	Year	Method	Dataset	Accuracy
Lo et al. [22]	2022	HyDL-IDS	$\mathcal{O}_\alpha$	97.11
Khan et al. [23]	2022	Multi-Stage IDS	$\mathcal{O}_\alpha$	98.10
Haddaji et al. [6]	2024	FL-TL	$\mathcal{O}_\alpha$	94.17
Alsaedi et al. [14]	2020	CART	$\mathcal{O}_\beta$	88.00
Zhang et al. [5]	2024	HPPELM	$\mathcal{O}_\gamma$	94.67
Proposed TIMIF	2024	APE-TIBD	$\mathcal{O}_\alpha$	99.98
			$\mathcal{O}_\beta$	99.98
			$\mathcal{O}_\gamma$	98.65
		APE-TIATI	$\mathcal{O}_\alpha$	99.97
			$\mathcal{O}_\beta$	99.20
			$\mathcal{O}_\gamma$	99.02

in values for DR, PR and F1 score with both datasets. These results confirm that proposed APE-TIBD and APE-TIATI module has performed well compared to existing baseline approaches in detecting threats.

E. Comparison with recent Techniques

Finally, we compare the TIMIF framework to extant state threat intelligence techniques to see how well it performs. The comparison of performance under the accuracy parameter is shown in Table VII. Most recent studies [15], [12], [11], [17], [16] have either failed to leverage openly accessible datasets or have used outdated datasets such as power system and UNSW-NB15 datasets, which do not involve new attacks and so have limited operational use for an ITS. Three different open source datasets, $\mathcal{O}_\alpha$, $\mathcal{O}_\beta$ and $\mathcal{O}_\gamma$ are used to assess the proposed TIMIF architecture. Moreover, we see that the APE-TIBD and APE-TIATI module of TIMIF framework has achieved higher accuracy compared to other approaches [22], [23], [6], [14] and [5]. The proposed TIMIF framework introduces a novel approach to threat intelligence in the VRCS. However, a notable challenge arises from the framework's inherent complexity, particularly its "black box" nature. This aspect significantly complicates efforts to understand and interpret how the TIMIF framework operates internally. As a result, analyzing the decision-making process that underlies the detection of cyber threats becomes a daunting task. Stakeholders and users may find it challenging to trace how specific detections were made or to validate the reasoning behind the identification of certain cyber threats. This limitation not only affects trust in the system's outputs but also hinders the ability for in-depth analysis and refinement of the threat detection mechanisms within the VRCS context.

VI. Conclusion and Future Work

In this article, we proposed TIMIF, a new threat intelligence modeling and identification framework for detecting and identifying threat types by analysing the network data from edge servers. TIMIF takes both RSUs and OBUs resource constraints and threat hunting performance requirements into consideration, and incorporates three DL-based key design modules, namely APE, TIBD and TIATI, to improve the security performance of VRCS network. The proposed TIMIF framework was evaluated using three datasets. Specifically, TIMIF achieved accuracy and detection rate close to 99% using HCRL-car hacking, ToN-IoT and CICIDS-2017 datasets. Experimental results showed that the TIMIF framework can significantly detect and identify recent attack patterns in VRCS network compared to some baseline and recent TI techniques. For the future work, we plan to develop a working prototype of TIMIF framework in actual VRCS environment to check its effectiveness. Another possibility is to extend TIMIF framework to address other relevant large data problems like image segmentation and object identification.

References

[1] C. Wu, Z. Cai, Y. He, and X. Lu, "A review of vehicle group intelligence in a connected environment," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 1865–1889, 2024.

[2] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibañez, "Internet of vehicles: architecture, protocols, and security," *IEEE internet of things Journal*, vol. 5, no. 5, pp. 3701–3709, 2017.

[3] S. Sharma and B. Kaushik, "A survey on internet of vehicles: Applications, security issues & solutions," *Vehicular Communications*, vol. 20, p. 100182, 2019.

[4] T. S. Darwish and K. A. Bakar, "Fog based intelligent transportation big data analytics in the internet of vehicles environment: motivations, architecture, challenges, and critical issues," *IEEE Access*, vol. 6, pp. 15 679–15 701, 2018.

[5] K. Zhang, J. Yang, Y. Shao, L. Hu, W. Ou, W. Han, and Q. Zhang, "Intrusion detection model for internet of vehicles using gripca and owelm," *IEEE Access*, 2024.

[6] A. Haddaji, S. Ayed, and L. C. Fourati, "A novel and efficient framework for in-vehicle security enforcement," *Ad Hoc Networks*, p. 103481, 2024.

[7] L. Nie, Z. Ning, X. Wang, X. Hu, J. Cheng, and Y. Li, "Data-driven intrusion detection for intelligent internet of vehicles: A deep convolutional neural network-based method," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2219–2230, 2020.

[8] L. Yang, A. Moubayed, and A. Shami, "Mth-ids: A multitiered hybrid intrusion detection system for internet of vehicles," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 616–632, 2022.

[9] B. Lampe and W. Meng, "Intrusion detection in the automotive domain: A comprehensive review," *IEEE Communications Surveys Tutorials*, vol. 25, no. 4, pp. 2356–2426, 2023.

[10] M. Al-Hawawreh, N. Moustafa, S. Garg, and M. S. Hossain, "Deep learning-enabled threat intelligence scheme in the internet of things networks," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2020.

[11] P. Koloveas, T. Chantzios, S. Alevizopoulou, S. Skiadopoulos, and C. Tryfonopoulos, "intime: A machine learning-based framework for gathering and leveraging web data to cyber-threat intelligence," *Electronics*, vol. 10, no. 7, p. 818, 2021.

[12] M. Kadoguchi, S. Hayashi, M. Hashimoto, and A. Otsuka, "Exploring the dark web for cyber threat intelligence using machine leaning," in *2019 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE, 2019, pp. 200–202.

[13] 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.

[14] 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.

[15] N. Moustafa, E. Adi, B. Turnbull, and J. Hu, "A new threat intelligence scheme for safeguarding industry 4.0 systems," *IEEE Access*, vol. 6, pp. 32 910–32 924, 2018.

[16] V. Atluri and J. Horne, "A machine learning based threat intelligence framework for industrial control system network traffic indicators of compromise," in *SoutheastCon 2021*. IEEE, 2021, pp. 1–5.

[17] N. Usman, S. Usman, F. Khan, M. A. Jan, A. Sajid, M. Alazab, and P. Watters, "Intelligent dynamic malware detection using machine learning in ip reputation for forensics data analytics," *Future Generation Computer Systems*, vol. 118, pp. 124–141, 2021.

[18] U. Noor, Z. Anwar, A. W. Malik, S. Khan, and S. Saleem, "A machine learning framework for investigating data breaches based on semantic analysis of adversary's attack patterns in threat intelligence repositories," *Future Generation Computer Systems*, vol. 95, pp. 467–487, 2019.

[19] H. K. KIM, "Car hacking dataset," 2018, online; accessed 1-Feb-2024. [Online]. Available: https://ocslab.hksecurity.net/Datasets/CAN-intrusion-dataset

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

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

[22] W. Lo, H. Alqahtani, K. Thakur, A. Almadhor, S. Chander, and G. Kumar, "A hybrid deep learning based intrusion detection system using spatial-temporal representation of in-vehicle network traffic," *Vehicular Communications*, vol. 35, p. 100471, 2022.

[23] I. A. Khan, N. Moustafa, D. Pi, W. Haider, B. Li, and A. Jolfaei, "An enhanced multi-stage deep learning framework for detecting malicious activities from autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25 469–25 478, 2022.

[24] Y. Yang, K. Zheng, B. Wu, Y. Yang, and X. Wang, "Network intrusion detection based on supervised adversarial variational auto-encoder with regularization," *IEEE Access*, vol. 8, pp. 42 169–42 184, 2020.

[25] S. Priya and R. A. Uthra, "An effective deep learning-based variational autoencoder for zero-day attack detection model," in *Inventive Systems and Control*. Springer, 2021, pp. 205–212.

[26] M. Tan, A. Iacovazzi, N.-M. M. Cheung, and Y. Elovici, "A neural attention model for real-time network intrusion detection," in *2019 IEEE 44th Conference on Local Computer Networks (LCN)*. IEEE, 2019, pp. 291–299.

[27] J. Ling, Z. Zhu, Y. Luo, and H. Wang, "An intrusion detection method for industrial control systems based on bidirectional simple recurrent unit," *Computers & Electrical Engineering*, vol. 91, p. 107049, 2021.

[28] C. Liu, Y. Liu, Y. Yan, and J. Wang, "An intrusion detection model with hierarchical attention mechanism," *IEEE Access*, vol. 8, pp. 67 542–67 554, 2020.

[29] P. Kumar, G. P. Gupta, and R. Tripathi, "A distributed ensemble design based intrusion detection system using fog computing to protect the internet of things networks," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–18, 2021.