A Rule-based Intrusion Detection System for NDN-based VANETs

Sruthi Cheriyachan Veettili[a,1], Muhammad Atif Ur Rehman [a,2], Waqas Khalid[b], Mohammed Al-Khalidi[a,3], Byung Seo Kim[c]

[a]*Department of Computing and Mathematics, Manchester Metropolitan University,* Manchester, UK.
[1]sruthicv92@gmail.com, [2]m.atif.ur.rehman@mmu.ac.uk, [3]m.al-khalidi@mmu.ac.uk
[b]*Institute of Industrial Technology*, *Korea University,* Sejong, South Korea, waqas283@korea.ac.kr
[c]*Department of Communication and Software Engineering*, *Hongik University,* Sejong, South Korea, jsnbs@hongik.ac.kr

Abstract—Vehicular Ad-Hoc Networks (VANETs), a specialized type of Mobile Ad-Hoc Networks, facilitate critical communications between vehicles such as vehicle-to-vehicle and vehicle-to-roadside infrastructures. Owing to the transmission of critical safety messages, the VANETs are prone to various cyberattacks aimed at compromising information such as safety message authenticity. To address VANETs' inherent need for robust safety message authentication, which traditional protocols often lack, Named Data Networking (NDN) appears as an alternative networking architecture, offering enhanced Data packet-level security. Utilizing NDN, this paper proposes a novel Intrusion Detection System (IDS) for VANETs with the precise aim of validating the authenticity of safety messages. To this end, the proposed scheme employs a rule-based approach leveraging vehicular sensor data to determine the integrity of received safety messages. The proposed IDS scheme is implemented and tested in a Python-based NDN-VANET simulation environment and demonstrates a minimum accuracy rate of 95% across various safety message categories.

***Keywords—VANETs, Cyber Security, NDN, Safety Message, Intrusion Detection System*.**

I. INTRODUCTION

In an increasingly interconnected world, modern vehicles going beyond their traditional role i.e., the mode of transportation, are transforming into integral computation and communication nodes within a vast network aiming to improve road safety, traffic management, and driving experience. This progression is pushed by the communication framework: Vehicular Ad-Hoc Networks (VANETs) [1], with the primary objective of fostering effective and safe transportation. VANETs facilitate various message types, including infotainment, safety, and transport efficiency messages. Among them, the safety messages are paramount due to their potential impact on drivers, passengers and pedestrians' lives if compromised. Owing to VANETs' high mobility and intermittent connectivity, conventional in-efficient TCP/IP protocols lack in providing a seamless and secure communication framework, necessitating a more secure, content-centric networking architecture. The Named Data Networking (NDN), has appeared as an optimal solution, matching the content-centric nature of message flow in VANETs, security requirements and connectivity constraints.

One major paradigm shift in NDN is securing the content itself while both in transit and when stored on a node in contrast to securing the communication channels such as in the TCP/IP, mitigating common TCP/IP attacks such as spoofing. This paradigm shift prioritizes secure content-based transmission and enables content dissemination to all available network interfaces simultaneously. Moreover, in contrast to TCP/IP, NDN provides various other unique features, for instance, 1) hierarchal and semantically meaningful names: by utilizing hierarchical and semantically meaningful names (similar to the application layer URLs) to route the packet on the network layer, NDN enables similar request (Interest) aggregation and response (Content) multicasting, 2) simplified mobility: enabling content requests by hierarchically and semantically meaningful names, irrespective of physical location, as opposed to the complex node mobility management across networks in TCP/IP, and 3) efficient caching: allowing content caching closer to consumers (end users' requesting the content), reducing server load and enhancing network efficiency.

In addition to the aforementioned unique and captivating features, NDN is also the most suitable communication architecture for VANETs due to the inherent content-centric nature of vehicular applications, prioritizing access to content over the specific physical location of the nodes i.e., vehicles, roadside units (RSUs), or cloud servers storing the content.

Although the NDN resolves many issues that TCP/IP-based address-centric VANETs confront, various challenges nevertheless persist and require novel solutions. For instance, the integrity of safety messages in VANETs is critical due to their potential to cause dangerous situations, even accidents, endangering human lives [2]. In NDN architecture, verifying the integrity of incoming messages in real-time is challenging, as the digital signature fetching process using the name in the key locator field and forwarding requests to the cloud server may consume excessive time, resulting in dropping the packet. Moreover, the exclusive reliance on the vehicle's digital signatures to verify the content could lead to hazardous consequences, specifically in the scenarios when the vehicle itself is compromised.

To resolve these issues, this paper proposes a rule-based intrusion detection system (IDS) that has the potential to efficiently determine the authenticity of safety messages in real time, ensuring vehicle reliance on safety messages in critical scenarios. The following are the core contributions of the papers.

- A rule-based IDS scheme is proposed which takes into account the various vehicle sensor parameters and predefined threshold values and evaluates the authenticity of the safety message employing the proposed algorithm. The proposed algorithms determine if the received message is authentic. In case, the message is spurious and potentially generated by the malicious node, the receiving node drops the message avoiding hazardous situations and improving network performance.
- A Python-based NDN simulator development incorporating all required IDS implementation such as integration of sensor values for assessing safety

message integrity, and basic NDN communication implementation such as Interest/Data messages generation and transmission in the network.

The rest of the paper is organized as follows. Section II presents a nutshell overview of NDN. Section III provides a brief literature review of various IDS and message authentication schemes in NDN. Section IV presents the proposed IDS and its algorithmic framework. Section V covers the proposed scheme evaluation details. Section VI outlines the limitations of the proposed scheme and possible future research directions, and finally, the paper concludes in Section VII.

II. NDN: In a Nutshell

The NDN is a paradigm shift from an in-efficient address-centric communication model to a data/content-centric approach, emphasising retrieving content directly based on its hierarchically and semantically meaningful names rather than relying on the particular IP address [3]. To this end, the communication nodes such as vehicles, RSUs, routers and cloud servers utilising NDN architecture are equipped with three key data structures to facilitate the NDN functionalities.

Content Store (CS): The CS is employed to store incoming Content/Data Packets embedded with digital signatures, serving multiple consumer requests in future for the similar content.

Pending Interest Table (PIT): The PIT keeps track of the Interest packets arriving from downstream nodes, aiding the corresponding Content/Data packet delivery. The PIT's major advantage is an aggregation of similar incoming Interest messages, reducing the redundant upward network traffic transmission.

Forwarding Information Base (FIB): The FIB which is usually populated by the routing information based (RIB) which in turn is managed by different routing protocols, is employed to determine the next hops for upstream Interest Packet forwarding.

Fig. 1 shows the overall NDN communication process. The communication is initiated when the consumer node forwards an Interest Packet, comprising the data/content name, nonce, and interest lifetime, among other key fields. Each intermediate node performs the following tasks by utilising the above-mentioned CS, PIT and FIB data structures.

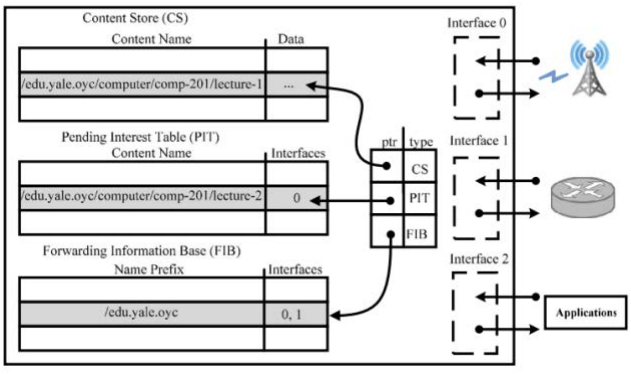

Fig. 1. A common NDN node architecture

Upon receiving Interest requests, the node first searches the PIT to check if a similar request is already in the queue. If present, the request is stored by aggregating the incoming interface ID in the face list. In the absence of a PIT entry, the node checks the CS for content availability. If present, the content is retrieved and forwarded to the downstream nodes. In case of data unavailability, both in PIT and CS, a new PIT entry is created, and the Interest packet is forwarded to the upstream node using the appropriate interface fetched from FIB. Once the Interest packet reaches the producer node (host), the signed content is returned all the back to the consumer by following the breadcrumb path created by the PIT entries. As the packet traverses back to the consumer, all the PIT entries are purged, creating space for future entries. The structure of Interest and Data Packets are depicted in Fig. 2.

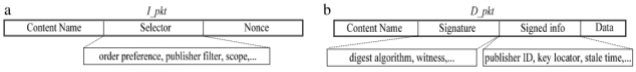

Fig. 2. Interest Packet and Data Packet in NDN

III. Related Work

This section critically reviews the most recent advancements in NDN and VANETs in terms of message authentication and IDS. To this end, this section is further divided into three subsections 1) IDS specifically designed for VANETs, 2) IDS tailored for NDN environments, and 3) message authentication mechanisms within NDN-based VANETs.

A. Intrusion Detection System for VANETs

The IDS for VANETs are primarily focused on addressing unique security challenges and threats inherent to these networks, with a key objective of ensuring seamless communication.

In [4], the authors introduce an IDS designed to resolve distributed denial of service (DDoS) attacks in VANETs. This system comprises two principal components: real-time network traffic collection using a micro-batch data processing model for enhanced performance, and a traffic detection module employing a Random Forest classification algorithm. Although the proposed scheme effectively resolves the DDoS issues in VANET, the proposed method consumes significant CPU space and memory because of the dependency on multiple external tools. In another scheme [5], the authors proposed a stream position performance analysis model (SPPA), which specifically aims at effectively detecting DDoS attacks. SPPA operates in three phases: cluster head selection, focusing on choosing the most robust node, stream position and cluster channel allocation (CCA) and attack detection. This method classifies neighbouring nodes into suspect, attacker, and normal groups, excluding attackers from cluster head elections. The SPPA also effectively detects the DDoS attack, however, at the cost of increased network traffic. Nandy et al. [6] propose a technique to identify malicious activities using a score table maintained by each vehicle about its neighbours, to analyse network patterns. Vehicles monitor real-time traffic, with a k-nearest neighbours (kNN) classifier employed for anomaly detection. The major drawback of this method is that it lacks experimental validation for real-time applicability.

B. *Intrusion Detection System for NDN*

IDS for NDN primarily focuses on addressing cyber-attacks that exploit the unique features of NDN architecture, particularly targeting content verification, caching, and data packet integrity.

A secure NDN architecture is proposed in [7] that integrates link layer protocols such as CAN, LIN, and automotive Ethernet into a cohesive and secure communication framework. The proposed approach comprises a three-step validation process for data packets, focusing on signature verification using methods similar to HMAC, and SHA256 with RSA. Although CPU utilisation is managed effectively, the proposed approach may lead to a high volume of unsatisfied interests. In [8], Xiong et al. propose a trigger-oriented countermeasure against timing-based attacks on caching in which the attacker deduces the status of cached data packet i.e., whether the content is cached. The proposed scheme detects the attack based on the recurrent pattern of attack monitored over a threshold number of occurrences and tracked using a data structure. Although the proposed scheme effectively detects the attacks, the naïve random delay-based scheme does not completely eradicate the timing attack. In another scheme [10] the authors propose an attention-based Long Short-Term Memory (LSTM) model for efficient detection of Interest Flooding Attacks (IFA). The proposed method utilises Hellinger distance [9] to recognise malicious prefixes in IFA and divide the whole process into defence, detection, and mitigation mechanisms. Although the response and mitigation modules work efficiently, the proposed approach demonstrates limitations in addressing low-rate IFA and other types of attacks.

C. *Message Authentication in NDN-based VANETs*

The safety validation techniques in NDN-based VANETs mainly rely on machine learning classifiers and encryption techniques.

The authors in [11] introduced a novel classification framework that employs machine learning algorithms such as KNN and logistic regression to identify malicious vehicles and fake messages in NDN-based VANETs. The proposed framework utilises stacking ensemble learning and implements the ML algorithms on a shared database comprising network traffic data collected from vehicles at neighbouring RSUs. Although the proposed scheme utilises four classifiers and yields promising results, the required CPU processing power and memory space required for training hinders the practical implementation of the proposed scheme. The authors in another scheme [12] exploit a concept based on the Swift Trust model [13], facilitating trust-based request-response communication, particularly for short-term trust decisions among vehicles. In this model, vehicles forward Basic Safety Messages (BSMs) every 0.1 seconds, comprising information such as velocity, brake status, acceleration, and steering angle. Although the proposed scheme provides solutions to mitigate certain attacks such as limiting interest packets to prevent DoS attacks and employing timestamps with nonces to counter replay attacks, the absence of experimental evaluation lacks in proving the effectiveness of the proposed schemes.

How proposed scheme is different? In contrast to IDS schemes discussed above that primarily rely on network parameters for assessing message integrity, the proposed IDS leverages the data from sensors installed within the vehicles. Modern vehicles are equipped with various sensors, continuously generating detailed information such as speed, engine RPM, brake usage, and GPS coordinates at millisecond intervals. This sensor data, stored in the vehicle's Electronic Control Unit, offer more accurate and reliable information to validate the integrity of received safety messages, thereby enhancing the effectiveness of the IDS in real-time vehicular environments.

IV. Proposed Intrusion Detection System

The proposed IDS architecture follows the RA-NDN framework proposed in [13] and is depicted in Fig. 3.

To elucidate further, let's consider an example in which a producer vehicle is moving on the road at a very high speed and crashes with another vehicle. As soon as the crash occurs the producer vehicle forwards a distress (safety) Data message in its vicinity using the NDN Data packet in TLV format on its wireless ad-hoc interface. The roadside unit (RSU) acting as a router in the close vicinity of a producer vehicle receives the Safety Data packet on its wireless ad-hoc interface. To this end, the proposed scheme enables unsolicited Data packet reception in the NDN. After receiving the Data packet, the RSU performs two operations: 1) Packet Inspection to identify if the received packet is a regular Data packet or safety Data packet, and 2) Intrusion Detection to recognise if the received packet is genuine or fake. The following subsections shed light on the aforementioned operations in detail.

Packet Inspection: To differentiate between the safety message and the conventional Data message, the proposed scheme utilises the TLV-based field. The packet-type field value 0 represents a conventional Data packet, whereas the other values represent the safety/distress Data packet according to Table 1. Upon receiving the Data packet, the RSU invokes the PacketInspection function. The PacketInspection function decapsulates the Data packet and extracts the packet-type field. If the received packet is a conventional Data packet the PacketInspection function

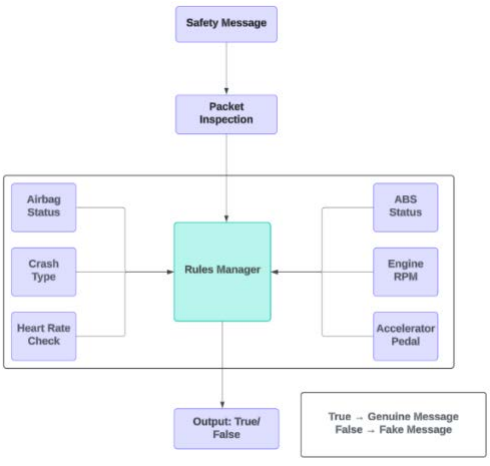

Fig. 3. Proposed IDS method.

invokes the conventional Data packet processing pipeline. However, if the received packet is a safety/distress packet,

the PacketInspection function invokes the proposed IDS functionality described below.

Intrusion Detection System: To identify if the received Data packet is authentic, the proposed scheme employs rule prioritisation techniques using Rules Manager. Upon receiving a Data packet, the Rules Manager extracts sensor values from the content field of a Data Packet. These sensor values include 1) airbag status, 2) anti-lock braking system status, 3) crash type, 4) engine revolutions per minute (RPM), and 5) heart rate.

Each of these sensor values is associated with a predefined rule. However, the sensor values may have different significance in determining authenticity. Therefore, a prioritisation mechanism is paramount to accurately identify the significance of a message. For instance, the airbag status might be given higher priority considering the direct correlation with crash authenticity. On the contrary, the engine RPM, although significant, may receive a lower priority due to its lower association with crash scenarios. Following this, the proposed IDS system using Rules Manager then compares these prioritised sensor values against their respective thresholds. As a result, the outcome of each rule is influenced by its assigned priority, ensuring that more crucial sensors have a greater impact on the authenticity decision. If the aggregated result of these weighted rule evaluations meets the criteria set by the Rules Manager, the Data packet is classified as 'authentic' and subsequently, RSUs forward the Data packet to neighbouring vehicles to take precautionary measures to avoid further accidents. The Data packet is dropped if the rules manager IDS criteria classify it as 'fake'.

Rules Manager Decision Criterion: The proposed IDS primarily focuses on safety message transmission. Table 1 outlines a rule set against which each incoming Data Packet is evaluated whereas corresponding algorithms 1-7 enforce these rules on the received Data packet.

TABLE 1: Rules for safety messages

Serial No.	Safety Message	Definition	Algorithms
1	EEBL	Electronic Emergency Brake Light	2,3,4,6
2	PCN	Post Crash Notification	3,4,5,6
3	ESA	Emergency Services Alert	3,4,5
4	MA	Medical Assistance	3,5
5	EN	Evacuation Notice	3,5,7,8

Algorithm 1 blends and validates rules criteria by invoking all algorithms i.e., from algorithms 2 to 8 to verify the authenticity of the received packet.

ALGORITHM 1: IDS FOR SAFETY MESSAGE	
	***Input**: Vehicle data, sensor data, content store*
	***Output**: bool*
1	**IDS_Value** ← *Representing Intrusion Detection System for Safety Message*
2	**if** tlv type in [EEBL, PCN, ESA, MA, EN] **then**
3	***HeartRate*** ← GetCurr_HeartRate () // *Algorithm 4*
4	**if** tlv type in [PCN, ESA, MA] **then**
5	***AirBag*** ← GetAirbagStatus () // *Algorithm 2*
6	**if** tlv type in [EEBL, PCN, MA] **then**
7	***TypeofCrash*** ← GetCrashType () // *Algorithm 6*
8	**if** tlv type in [EEBL, PCN] **then**
9	***AbsValue*** ← GetAbsStatus() // *Algorithm 3*
10	**if** tlv type in [PCN] **then**
11	***EngineRPM*** ← GetEngineRPM () // *Algorithm 5*
12	**if** tlv type in [PCN, MA, EN] **then**
13	***Evac_Status*** ← GetEvacStatus () // *Algorithm 8*
14	**IDS_Value** ← *HeartRate + AirBag + TypeofCrash + AbsValue + EngineRPM + Evac_Status*
15	$\mathbf{IDS}_{weight}$ ← $airbag_{priority}$ + $RPM_{priority}$ + $heartrate_{priority}$ + $ABS_{priority}$ + $crashtype_{priority}$ + $accpedal_{priority}$
16	**If** (**IDS_Value** == 'true') & ($\mathbf{IDS}_{weight}$ > 0.7) **then** // *Confirms that the message is genuine*
17	***Return*** **IDS_Value**

Algorithm 2 verifies airbag deployment which is typically essential in post-crash safety messages as the airbag eruption signifies the probability that the vehicle has crashed.

ALGORITHM 2 CHECK AIRBAG_STATUS	
	***Input**: data*
	***Output**: [status, priority]*
1	***Air_Bag*** ← *Representing the current value of airbag deployment.*
2	*AirBag* ← GetAirbagStatus ()
3	**If** (*AirBag == 0)* **then**
4	***Air_Bag*** = false
5	**else**
6	***Air_Bag*** = true
7	$airbag_{priority}$ =0.4
8	**Return** ***[Air_Bag,*** $\mathbf{airbag}_{priority}$***]***

Algorithm 3 validates if the ABS has been activated.

ALGORITHM 3 CHECK ABS_STATUS	
	***Input**: Vehicle data*
	***Output**: [status, priority]*
1	***ABS*** ← *Representing the current value of Anti-lock braking system applied.*
2	*AbsValue* ← GetAbsStatus ()
3	**If** (*AbsValue == 0)* **then**
4	**ABS** = false
5	**else**
6	**ABS** = true
7	$ABS_{priority}$ = 0.1
8	**Return** ***[ABS,*** $\mathbf{ABS}_{priority}$ ***]***

Algorithm 4 considers heart rate, comparing historical values obtained from the content store with the current heart rate to detect the possibility of a crash. The normal heart range is considered between 80 to 100 beats per minute.

ALGORITHM 4 CHECK HEART RATE		
	Input: *Vehicle data, Content Store*	
	Output: *[status, priority]*	
1	***Avg_HeartRate*** ← *Represent previous heart rates stored in Content Store*	
2	***Curr_HeartRate*** ← *Represent the current heart rate of the driver*	
3	***Valid_HRate*** ← *Represent Valid heart rate for the crash scenario.*	
4	***HeartRate*** ← GetCurr_HeartRate ()	
5	***AvgHeartRate*** ← GetAvg_HeartRate ()	
6	**If** (***HeartRate*** > ***AvgHeartRate***) *and* (***HeartRate*** > *100)* **then** *// AvgHeartRate is the average of al the heart rates stored from the beginning of the journey*	
7		***Valid_HRate*** = true *// Valid case for crash*
8		$heartrate_{priority}$ =0.2
9	**else**	
10		***Valid_HRate*** = false
11	**Return** ***[Valid_HRate, heartrate$_{priority}$]***	

Algorithm 5 validates engine RPM, which typically exceeds 2000 in crash scenarios [15].

ALGORITHM 5 CHECK ENGINERPM		
	Input: *Vehicle data*	
	Output: *[status, priority]*	
1	***Eng_RPM*** ← *Representing the current value of engine rpm (revolutions per minute).*	
2	*EngineRPM* ← GetEngineRPM ()	
3	**If** (*EngineRPM > 2000)* **then**	
4		***Eng_RPM*** = true *// Valid case for crash*
5		$RPM_{priority}$ =0.2
6	**else**	
7		***Eng_RPM*** = false
8	**Return** ***[Eng_RPM, RPM$_{priority}$]***	

Algorithm 6 identifies the crash type which can classified as 'Front', 'Side', or 'Rear'.

ALGORITHM 6 CHECK CRASH TYPE		
	Input: *Vehicle data*	
	Output: *[status, priority]*	
1	***Crash_Type*** ← *Determines which type of crash had happened from three possible crash types.*	
2	*TypeofCrash* ← GetCrashType ()	
3	**If** (*TypeofCrash == 'Front'|'Rear'|'Side')* **then**	
4		***Crash_Type*** = true *// Valid case for crash*
5		$crashtype_{priority}$ =0.05
6	**else**	
7		***Crash_Type*** = false
8	**Return** ***[Crash_Type, crashtype$_{priority}$]***	

Finally, Algorithm 7 evaluates the likelihood of an explosion following a crash; if criteria are met, the IDS triggers an evacuation message to all neighbouring vehicles.

ALGORITHM 7 EVACUATION		
	Input: *Vehicle data*	
	Output: *[status, priority]*	
1	***Evac_Status*** ← *Determines the chances of explosion because of crash and if evacuation needed.*	
2	***Speed*** ← GetVehicleSpeed ()	
3	***AirBag*** ← GetAirbagStatus () //Algorithm 2	
4	***TypeofCrash*** ← GetCrashType () // Algorithm 6	
5	**If** (***Speed*** *> 180)* ***and*** *(**AirBag*** = =1) ***and*** (***TypeofCrash*** == *'Front'|'Rear'|'Side')* ***then***	
6		***Evac_Status*** = true
7	**else**	
8		***Evac_Status*** = false
9	**Return** ***[Evac_Status, crashtype$_{priority}$]***	

V. PERFORMANCE EVALUATION AND DISCUSSION

Simulation Setup: To validate the effectiveness of the proposed IDS scheme, an NDN-based communication framework using a Python programming language is initially developed. The NDN core features such as consumer application, producer application, Interest packet structure, Data packet structure, interface implementation, and core forwarding daemon implementation, among others are developed from scratch, fulfilling the minimum viable NDN communication requirements. Subsequently, the proposed IDS Rules Manager is implemented and integrated into producer and consumer applications within RSU to validate the received message authenticity. Notably, we opted against utilising ndnSIM, a widely used NDN simulator due to its limitation in supporting machine learning algorithms, which are planned for future intelligent IDS implementation. This necessitates the utilisation of Python owing to its extensive compatibility with a wide range of machine-learning algorithms.

To rigorously validate the efficacy of the proposed IDS system, 250 messages are transmitted including both genuine and fake embedded with sensor values such as speed, latitude, longitude, heart rate, airbag status, ABS activity, RPM, pedal usage, crash type, and timestamp. These messages include 'EEBL', 'PCN', 'ESA', and other types, and are over 5 seconds defined period. Table 1 outlines a detailed overview of various safety messages, their definitions, and the corresponding validation rules.

Results & Discussion: Fig. 4 presents results under various test scenarios for different types of safety messages. As demonstrated, the proposed IDS classified 47 as genuine and 203 as fake out of a total of 250 EEBL messages, achieving 96% accuracy. For PCN, the IDS classified 43 as genuine and 207 as fake, achieving 97% accuracy. A substantially higher number of fake messages are included in the case of PCN because of their critical impact on passenger safety. Similar testing is conducted for other message types such as ESA, MA, and EN, yielding 96%, 95%, and 95% accuracy respectively. The graphical representation combining all these message-type results (by taking the average) is presented in Fig. 5.

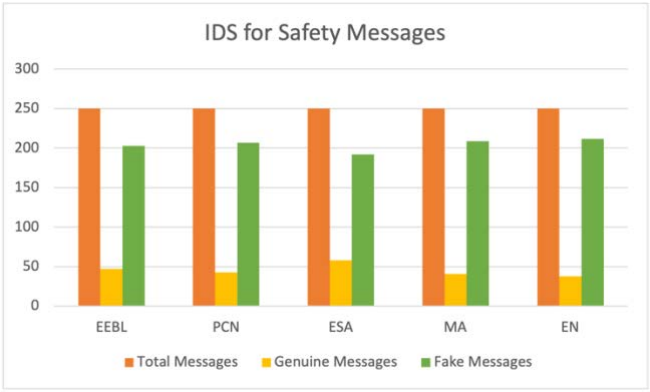

Fig. 4: Genuine and fake message validation

Fig. 5: Average of all Genuine and fake message validation

VI. Future Work

Although the proposed IDS accurately classify authentic and fake messages, a plethora of issues still require further research effort in this domain. At first, we acknowledge the absence of a benchmark dataset for testing. The dummy data, created for rule set verification, may not precisely replicate the sensor values extracted from vehicles in real-world distress or accidents. Future work in this domain should delve into the effective and real-world retrieval of sensor values from the producer vehicles to identify which vehicular information is too sensitive to be disclosed to RSU. Moreover, currently, the use case scenario presented in this paper operates with a single Producer and RSU. However, real-life scenarios typically involve multiple producers, as accidents often occur between multiple vehicles. Therefore, enhancing the IDS for use across multiple vehicles and even to multiple RSUs is crucial as each RSU covers only a limited range. Another potential future work is the extensive comparison of the proposed scheme with existing IDS schemes proposed for NDN-based VANETs and with the traditional content integrity verification mechanism using the hierarchically and meaningful digital signature name in the key locator field, especially in the context of latency. Finally, as future work, we plan to utilise a machine learning algorithm to dynamically evaluate a range of various sensor values, aiming to develop a robust IDS system that withstands other than the statically predefined values.

VII. Conclusion

This paper presents a rule-based IDS specifically tailored for NDN-based VANETs, designed and implemented in a Python-based simulator, emulating the NDN communication environment. The core objective of the proposed IDS is to accurately distinguish between legitimate and fake messages transmitted by vehicles by utilising a set of predefined rules. The proposed scheme collects data from various in-vehicle sensors, such as airbag status, anti-lock braking system activity, heart rate, and crash type, to verify the integrity of safety messages in post-crash scenarios, which most of the existing IDS schemes in literature lack. The simulation analysis reveals promising results. The proposed IDS achieves a minimum accuracy of 95% across all safety message categories.

Acknowledgement

This research is supported in part by Manchester Metropolitan University, UK, in part by the Strategic Networking & Development Program funded by the Ministry of Science and ICT through the National Research Foundation of Korea (RS-2023-00277267), and in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) (NRF-2022R1I1A1A01071807).

References

[1] M. A. Al-shareeda, M. A. Alazzawi, M. Anbar, S. Manickam and A. K. Al-Ani, "A Comprehensive Survey on Vehicular Ad Hoc Networks (VANETs)," 2021 International Conference on Advanced Computer Applications (ACA), Maysan, Iraq, 2021, pp. 156-160, doi: 10.1109/ACA52198.2021.9626779.

[2] P. Manimaran and A. R. K. P., "NDNIDS: An Intrusion Detection System for NDN Based VANET," 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 2020, pp. 1-5, doi: 10.1109/VTC2020-Spring48590.2020.9129365.

[3] D. Saxena, V. Raychoudhury, N. Suri, C. Becker, and J. Cao, "Named Data Networking: A survey," *Computer Science Review*, vol. 19, pp. 15–55, Feb. 2016, doi: https://doi.org/10.1016/j.cosrev.2016.01.001.

[4] Gao, Y., Wu, H., Song, B., Jin, Y., Luo, X., & Zeng, X. (2019). A distributed network intrusion detection system for distributed denial of service attacks in vehicular ad hoc networks. *IEEE Access*, *7*, 154560–154571. https://doi.org/10.1109/ACCESS.2019.2948382

[5] R. Kolandaisamy *et al.*, "Retraction Note to: A stream position performance analysis model based on DDoS attack detection for cluster-based routing in VANET," *Journal of Ambient Intelligence and Humanized Computing*, Jun. 2022, doi: https://doi.org/10.1007/s12652-022-04213-0.

[6] T. Nandy, R. M. Noor, M. Yamani Idna Bin Idris and S. Bhattacharyya, "T-BCIDS: Trust-Based Collaborative Intrusion Detection System for VANET," 2020 National Conference on Emerging Trends on Sustainable Technology and Engineering Applications (NCETSTEA), Durgapur, India, 2020, pp. 1-5, doi: 10.1109/NCETSTEA48365.2020.9119934.

[7] Z. Tntech *et al.*, "Securing Automotive Architectures with Named Data Networking." Available: https://arxiv.org/pdf/2206.08278.pdf

[8] Xiong, W., IEEE Computer Society, International Association for Computer & Information Science, Pattern Recognition and Machine Intelligence Association., & Institute of Electrical and Electronics Engineers. (n.d.). *17th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2018) : proceedings : June 6-8, 2018, Singapore*.

[9] Basu, A., Mandal, A., & Pardo, L. (2010). Hypothesis testing for two discrete populations based on the Hellinger distance. *Statistics and Probability Letters*, *80*(3–4), 206–214. https://doi.org/10.1016/j.spl.2009.10.008

[10] Zhang, X., Li, R., & Hou, W. (2022). Attention-Based LSTM Model for IFA Detection in Named Data Networking. *Security and Communication Networks*, *2022*. https://doi.org/10.1155/2022/1812273

[11] Liang, J., Chen, J., Zhu, Y., & Yu, R. (2019). A novel Intrusion Detection System for Vehicular Ad Hoc Networks (VANETs) based on differences of traffic flow and position. *Applied Soft Computing Journal*, *75*, 712–727. https://doi.org/10.1016/j.asoc.2018.12.001

[12] S. K. Ramani and A. Afanasyev, "Rapid Establishment of Transient Trust for NDN-Based Vehicular Networks," 2020 IEEE International Conference on Communications Workshops (ICC Workshops), Dublin, Ireland, 2020, pp. 1-6

[13] D. Murthy, A. Rodriguez and J. Lewis, "Examining the Formation of Swift Trust within a Scientific Global Virtual Team," 2013 46th Hawaii International Conference on System Sciences, Wailea, HI, USA, 2013, pp. 353-362, doi: 10.1109/HICSS.2013.211.

[14] S. Tiennoy and C. Saivichit, "Using a Distributed Roadside Unit for the Data Dissemination Protocol in VANET With the Named Data Architecture," in IEEE Access, vol. 6, pp. 32612-32623, 2018, doi: 10.1109/ACCESS.2018.284008

[15] Team Ackodrive, "What is RPM in Cars?," AckoDrive, Jun. 07, 2022. https://ackodrive.com/car-guide/what-is-rpm/