Wireless Location Verification and Acquisition Using Machine Learning

Ihsan Ullah

A dissertation submitted to the Graduate Research School of
The University of New South Wales
in total fulfillment of the requirements for the degree of

Doctor of Philosophy

School of Electrical Engineering and Telecommunications
The University of New South Wales

July 2021

Thesis submission for the degree of Doctor of Philosophy

Thesis Title and Abstract | Declarations | Inclusion of Publications Statement | Corrected Thesis and Responses

Thesis Title

Wireless location verification and acquisition using machine learning

Thesis Abstract

Traditional wireless location verification (authentication) is only feasible under the assumption that radio propagation is described by simple time-independent mathematical models. A similar situation applies to location acquisition, albeit to a lesser extent. However, in real-world situations, channel conditions are rarely well-described by simple mathematical models. In this thesis, novel location verification and acquisition techniques that integrate machine learning algorithms into the decision process are designed, analysed, and tested. Through the use of both simulated and experimental data, it is shown how the novel solutions developed remain operational in unknown time-varying channel conditions, thus making them superior to existing solutions, and more importantly, deployable in real-world scenarios. Location verification will be of growing importance for a host of emerging wireless applications in which location information plays a pivotal role. The location verification solutions offered in this thesis are the first to be tested against experimental data and the first to invoke machine learning algorithms. As such, they likely form the foundation for all future verification algorithms.

Thesis submission for the degree of Doctor of Philosophy

Thesis Title and Abstract | Declarations | Inclusion of Publications Statement | Corrected Thesis and Responses

ORIGINALITY STATEMENT

☑ I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

COPYRIGHT STATEMENT

☑ I hereby grant the University of New South Wales or its agents a non-exclusive licence to archive and to make available (including to members of the public) my thesis or dissertation in whole or part in the University libraries in all forms of media, now or here after known. I acknowledge that I retain all intellectual property rights which subsist in my thesis or dissertation, such as copyright and patent rights, subject to applicable law. I also retain the right to use all or part of my thesis or dissertation in future works (such as articles or books).

For any substantial portions of copyright material used in this thesis, written permission for use has been obtained, or the copyright material is removed from the final public version of the thesis.

AUTHENTICITY STATEMENT

☑ I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis.

Thesis submission for the degree of Doctor of Philosophy

Thesis Title and Abstract | Declarations | **Inclusion of Publications Statement** | Corrected Thesis and Responses

UNSW is supportive of candidates publishing their research results during their candidature as detailed in the UNSW Thesis Examination Procedure.

Publications can be used in the candidate's thesis in lieu of a Chapter provided:

- The candidate contributed **greater than 50%** of the content in the publication and are the "primary author", i.e. they were responsible primarily for the planning, execution and preparation of the work for publication.
- The candidate has obtained approval to include the publication in their thesis in lieu of a Chapter from their Supervisor and Postgraduate Coordinator.
- The publication is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in the thesis.

☑ The candidate has declared that **some of the work described in their thesis has been published and has been documented in the relevant Chapters with acknowledgement**.

A short statement on where this work appears in the thesis and how this work is acknowledged within chapter/s:

> In Chapter 1 of the thesis, I have provided the details of how I have used my publications in various chapters of my thesis.

Candidate's Declaration

I declare that I have complied with the Thesis Examination Procedure.

Thesis/Dissertation Sheet

Surname/Family Name	:	**Ullah**
Given Name/s	:	**Ihsan**
Abbreviation for degree as give in the University calendar	:	**PhD**
Faculty	:	**Engineering**
School	:	**Electrical Engineering and Telecommunications**
Thesis Title	:	**Wireless Location Verification and Acquisition Using Machine Learning**

Abstract 350 words maximum: (PLEASE TYPE)

Traditional wireless location verification (authentication) is only feasible under the assumption that radio propagation is described by simple time-independent mathematical models. A similar situation applies to location acquisition, albeit to a lesser extent. However, in real-world situations, channel conditions are rarely well-described by simple mathematical models. In this thesis, novel location verification and acquisition techniques that integrate machine learning algorithms into the decision process are designed, analysed, and tested. Through the use of both simulated and experimental data, it is shown how the novel solutions developed remain operational in unknown time-varying channel conditions, thus making them superior to existing solutions, and more importantly, deployable in real-world scenarios. Location verification will be of growing importance for a host of emerging wireless applications in which location information plays a pivotal role. The location verification solutions offered in this thesis are the first to be tested against experimental data and the first to invoke machine learning algorithms. As such, they likely form the foundation for all future verification algorithms.

ORIGINALITY STATEMENT

'I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.'

Signed ……..

Date ……..

INCLUSION OF PUBLICATIONS STATEMENT

UNSW is supportive of candidates publishing their research results during their candidature as detailed in the UNSW Thesis Examination Procedure.

Publications can be used in their thesis in lieu of a Chapter if:

- The student contributed greater than 50% of the content in the publication and is the "primary author", ie. the student was responsible primarily for the planning, execution and preparation of the work for publication.
- The student has approval to include the publication in their thesis in lieu of a Chapter from their supervisor and Postgraduate Coordinator.
- The publication is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in the thesis.

Please indicate whether this thesis contains published material or not.

☐ *This thesis contains no publications, either published or submitted for publication*

☒ *Some of the work described in this thesis has been published and it has been documented in the relevant Chapters with acknowledgement*

☐ *This thesis has publications (either published or submitted for publication) incorporated into it in lieu of a chapter and the details are presented below*

CANDIDATE'S DECLARATION

I declare that:

- I have complied with the Thesis Examination Procedure
- where I have used a publication in lieu of a Chapter, the listed publication(s) below meet(s) the requirements to be included in the thesis.

Name	**Signature**	**Date (dd/mm/yy)**
Ihsan Ullah		

To my parents

Abstract

Emerging wireless communication networks face serious performance challenges due to an exponential rise in the number of connected devices and the limited options available to expand the spectrum bandwidths in the use of these networks. These performance challenges include, but are not limited to, coverage holes in the networks' footprint, capacity constraints due to spectrum limitations, and low end-user throughputs due to an increase in the interference levels and the number of connected devices.

The Internet of Things (IoT) will introduce billions of new devices to operate its numerous sub-applications (*i.e.*, smart home, smart health, Intelligent Transportation System (ITS), etc.), adding further to the challenges mentioned above. It has been revealed through continuous research that accurate location information of the devices can play a vital role in addressing the aforementioned performance challenges. Such information is usually provided to the network by the devices themselves. Acquiring location information is one aspect; more important is to verify this information before utilizing it for the network operation. Over the years, several information-theoretic and heuristic algorithms have been developed to verify the devices' reported location information. These algorithms, in general, have been designed for specific geographic and channel environments, hence have limitations in their operation. Moreover, they are not practical due to their inflexibility in tuning to the changing channel circumstances. This thesis considers state of the art Neural Network (NN) techniques to overcome such limitations and improve overall location verification performance. The location verification concepts discussed in this thesis are in the context of a vehicular ad-hoc network. Still, they are believed to apply to a wide range of other location-centric applications within 5G, IoT, and other wireless communication networks.

In the first half of this thesis, nineteen location acquisition algorithms are surveyed. Several notable location verification frameworks and their operational limitations are reviewed. An NN-based location acquisition algorithm is developed to acquire the location information of vehicles in the field. A deep study of the internal architecture of the developed NN-based location acquisition algorithm in conjunc-

tion with information-theoretic constructs is conducted. Next, another vital aspect of the acquired location information is addressed through NN, *i.e.*, its verification. An NN-based location verification framework is developed to address numerous limitations of the available non-NN-based location verification algorithms. Through simulated data, it is demonstrated how the working of the developed framework is free from numerous assumptions on the channel environment, which are required to operate non-NN-based location verification algorithms. The functioning of the designed NN-based location verification framework is further validated through real-world experimental data to show that such framework is realistic.

In the second half of this thesis, two key aspects that impact the overall performance of any NN framework are investigated. In particular, these aspects are investigated in the context of the designed location verification framework. The first key aspect is the training time of an NN. A derived information-theoretic bound is taken into account to streamline the training time of the designed NN-based location verification framework while ensuring that the framework's performance is not compromised. The second key aspect investigated is that a location verification NN with its parameters optimized under given channel specifications can perform to a threshold. To further enhance the location verification performance, a joint framework is formulated by combining two standalone location verification NNs. The joint framework improves the location verification performance beyond those of the standalone NNs.

Finally, this thesis is concluded by identifying new frontiers for the developed intelligent location verification framework.

List of Publications

During the course of this thesis project, a number of publications have been made based on the work presented here and are listed below for reference.

Journal Publication

1. **U. Ihsan**, S. Yan, and R. Malaney, "Location verification for emerging wireless vehicular networks," *IEEE Internet of Things journal*, vol. 6, no. 6, pp. 10261-10272, Aug. 2019.

Conference Publications

2. **U. Ihsan**, Z.Wang, R. Malaney, A. Dempster, and Shihao Yan, "Location verification performance in the presence of verifier location error," in *Proceedings of The International Global Navigation Satellite Systems (IGNSS2018) Conference*, Sydney, Australia, Feb 2018, pp. 1-8.

3. **U. Ihsan**, R. Malaney, and S. Yan, "Machine learning and location verification in vehicular networks," in *Proceedings of the 8th IEEE/CIC International Conference on Communications in China (ICCC2019)*, Changchun, China, Aug. 2019, pp. 91-95.

4. **U. Ihsan**, R. Malaney, and S. Yan, "Artificial intelligence and location verification in vehicular networks," in *Proceedings of the IEEE Global Communication Conference (GlobeCom)*, HI, USA, Dec. 2019, pp. 1-6.

5. **U. Ihsan**, R. Malaney, and S. Yan, "Neural network architecture and location estimation in the Internet of Things," in *Proceedings of the IEEE International Communication Conference (ICC)*, Montreal, QC, Canada, Jun. 2021, pp. 1-6.

6. **U. Ihsan**, and R. Malaney, "Combining different neural networks for location verification," Submitted to: *IEEE International Communication Conference (ICC)*.

Other Publication

7. S. Yan, **U. Ihsan**, and R. Malaney, "Location Verification for Future Wireless Vehicular Networks: Research Directions and Challenges," Accepted to appear in: *IEEE Network Magazine*.

Acknowledgment

I am truly grateful to Allah Subhanahu Wa Ta'ala for granting me the opportunity, strength, and ability to pursue and reach this far into my PhD.

I am ineffably indebted to my supervisor, Professor Robert A. Malaney, for his expertise, conscientious guidance, and encouragement. His passion for practical and world-class research has helped me transform my potentials into success. I am thankful to him for believing in me and providing me with the opportunity to work with him.

This dissertation is also an outcome of a wonderful collaboration with Dr Shihao Yan, who had the same passion and excitement for this research. This work would not have been possible without his stimulating discussions, suggestions, and encouragement.

I want to thank my parents from the bottom of my heart. Baba Gee and Ammi, you are physically no more with me, but I can spiritually feel your presence. You left me at a young age, but your legacy of 'Keep Working Hard' will stay with me forever.

Immense gratitude to my brother, Dr Ikram Ullah, for his constant guidance, encouragement, and much needed moral support throughout the PhD. I do not have words to thank you for always answering my calls at odd times whenever I felt stressed and lost.

Many thanks as always to my wife and children for their patience and support.

Thank you, UNSW, for providing me with all the means to complete this project. I was indeed lucky to have pursued my PhD in such a safe and beautiful environment. Life at UNSW was challenging yet enjoyable.

I was undoubtedly fortunate to have enjoyed the company of my friends Manzoor,

Alex, Waheeda, Saad, Jawwad, Haq, and Talal at the campus.

Last but not least, a debt of gratitude to this beautiful land of Australia, its people, and the government for providing me with all the valued support that helped me complete this research assignment.

Contents

List of Figures

Acronyms

AI	Artificial Intelligence
AoA	Angle of Arrival
BSs	Base Stations
CBSs	Covert Base Stations
CLs	Confidence Levels
CNNs	Convolutional Neural Networks
CRB	Cramer-Rao Bound
DRSS	Differential Received Signal Strength
FMBA	Fast Multi-hop Broadcast Algorithm
FNNs	Feedforward Neural Networks
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IoT	Internet of Things
ITS	Intelligent Transportation System
KL	Kullback–Leibler
LER-GR	Location Error Resilient Geographical Routing
LoS	Line-of-Sight
LRT	Likelihood Ratio Test
LSDS	Location Spoofing Detection System
LSTM	Long Short-Term Memory
LVSs	Location Verification Systems
M2M	Machine-to-Machine

MHLVP	Multihop Location Verification Protocol
ML	Machine Learning
mMIMO	massive Multiple-Input and Multiple-Output
MSE	Mean Square Error
MSR	Mutually Shared Region
MSRLV	Mutually Shared Region-based Location Verification
NLoS	Non-Line-of-Sight
NN-LVS	Neural Network-based Location Verification System
NNFs	Neural Network Frameworks
NNLEF	Nueral Network-based Location Estimation Framework
NNs	Neural Networks
PBS	Public Base Station
PC	Processing Center
ReLU	Rectifier Linear Unit
RNN	Recurrent Neural Network
RSS	Received Signal Strength
RSUs	Road Side Units
RTT	Round Trip Time
SNNF	Selective Neural Network Framework
SRVL	Simultaneous Reporting and Verification of Location
SVMs	Support Vector Machines
TAS	Trusted Autonomous Systems
TLB	Total Error Lower Bound
ToA	Time of Arrival
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VANETs	Vehicular Ad-hoc Networks

VC	Verification Centre
WAVE	Wireless Access in Vehicular Environments
WHO	World Health Organization
WNNF	Weighted Neural Network Framework

Chapter 1

Background

This chapter first highlights the significance of location information (especially its verification) for emerging wireless networks. This is followed by detailing the crucial role that such information can play in wireless vehicular networks. The integration of Machine Learning (ML) algorithms for Vehicular Ad-hoc Networks (VANETs) operation is then briefly discussed. Although the main focus of this thesis is location verification which is studied in the context of a VANET, a brief overview of a few notable positioning algorithms is presented for the interest of readers before discussing location verification in detail. Next, some of the notable works carried out in the field of location verification are covered. First, a few heuristic location verification frameworks are studied. Then a brief description of several notable information-theoretic location verification frameworks that use physical layer properties of the vehicles' transmitted signals is presented. The significance of ML algorithms for location verification is highlighted next. A summary of the basics of Neural Networks (NNs) and an explanation of their internal working is stated next. Finally, the main contributions of this thesis are detailed.

1.1 Introduction

Wireless communication has evolved over the years, transforming various aspects of society. A few decades ago, the services offered by the wireless communication networks (*i.e.*, 1G and 2G) were merely restricted to basic telephony, short message service, and end-user data throughput speeds in kilobits per second range. The current more advanced 5G and the Internet of Things (IoT) communication networks improve the end-user data throughput speed to gigabits per second range. Additionally, these networks extend services beyond basic telephony and offer massive Machine-to-Machine (M2M) communication. The M2M communication enables a wide range of applications within these networks. Examples are smart homes, smart industries, Trusted Autonomous Systems (TAS), VANETs, etc. 5G and the IoT would need to add billions of new devices to the existing users base for connectivity to enable such applications.

Moreover, these networks will demand seamless coverage and ultra-high data rates in addition to ultra-low latency. The wireless networks will require many new transceiver Base Stations (BSs) to be installed on top of the existing infrastructure to meet these demands. With the increased BSs' density, the inter-site distances can shrink to as low as approximately one hundred meters. With tighter frequency reuse and limited choices to expand the spectrum bandwidth in use for these networks, the networks will be at risk of increased interference levels. These increased interference levels will adversely affect the networks' coverage, capacity, latency, and end-user data throughput rates.

1.2 Network Problems and Location Information

Location information of the users and devices will play an ever-increasing role in the current and future wireless networks [1–7]. This information will be important in minimizing the networks' interference levels, henceforth in enhancing the coverage, capacity, latency, and end-user data throughput speeds. It can be stated that loca-

tion information will form the basis of many network decisions and play a key role in various aspects of the network operation, with safety more critical in emerging vehicular networks. However, such information is user-dependent. This means that the users usually collect their location information using a Global Positioning System (GPS) or any satellite-based positioning system and report it to the network once requested. The network then utilizes such reported location information to empower its numerous operations. A few examples of such operations are geographical routings [1–3], location-based access control [4, 5], and location-based services [6, 7].

There is a chance that a user may provide its wrong location information intentionally in an attempt to cheat the network. A possibility exists where the user may forward its false location information to the network due to some software/hardware issues or due to extreme weather conditions. A user, if able to spoof its location (a sample spoofing scenario is highlighted in Fig. 1.2), can produce serious security risks [1, 2, 4, 5, 8, 9]. Such risks are indisputably more severe in the case of VANETs [10–13]. It is, therefore, crucial to verify such location information before using it for the network operation.

This thesis considers a VANET as a system and discusses location verification concepts in the context of this system. This system's operation evolves around its users' reported location information. The location information of the users is reported to the system by the users themselves through wirelessly transmitted signals over the air interface. This thesis further complements the system's functionality by designing novel and intelligent solutions to verify the users' reported location information. The novel solutions consider the properties of the users' transmitted wireless signals for their operation in an attempt to streamline the overall system's performance. The same novel solutions are believed to be applicable to a wide range of location-centric applications within 5G/IoT (*i.e.*, beamforming, massive Multiple-Input and Multiple-Output (mMIMO), highway pooling, and smart parking etc.), and in the defence industry.

Many researchers in the past have developed several traditional as well as information-theoretic Location Verification Systems (LVSs). These LVSs can efficiently perform

if various constraints, *i.e.*, channel parameters and threat conditions etc., assumed at the time of design of such frameworks become a reality. In real-world situations, such assumptions are a poor representation of the actual physical system conditions. Moreover, it is challenging to know channel attributes, noise parameters, and other system descriptors in advance. Besides, channel and threat conditions can change on the go. For example, channel characteristics are different in rural areas than dense city centers; hence a traditional or information-theoretic LVS framework designed for one area may not function well in another area (with a different channel profile). Also, abrupt weather changes can influence the performance of such LVSs. To compensate for such real-world challenges and spatial as well as temporal characteristics of the system parameters, a more intelligent LVS beyond traditional or information-theoretic LVSs is required. I think ML would be the essential ingredient of such an intelligent LVS.

1.3 Location Verification in VANETs

A VANET is an example of an Intelligent Transportation System (ITS). A VANET enables real-time communication in both Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I). The working of a VANET relies on the communication between vehicles and other roadside infrastructure. The infrastructure includes static BSs installed at appropriate locations around the roads. These BSs are widely known as Road Side Units (RSUs). A central RSU, commonly known as the Processing Center (PC), collects and processes V2V and V2I communication before making critical network decisions. A model representing a VAENT with RSUs is shown in Fig. 1.1. A VANET can also function merely based on V2V communication, *i.e.*, without any assistance from the RSUs. In such a case, a master vehicle (with its location authenticated previously) will play the role of validating the locations reported by other vehicles. We can think of a police car playing the role of the master vehicle. A model of this latter type of a VANET is shown in Fig. 1.2.

VANETs improve location-based routing [14], minimize traffic congestion [15],

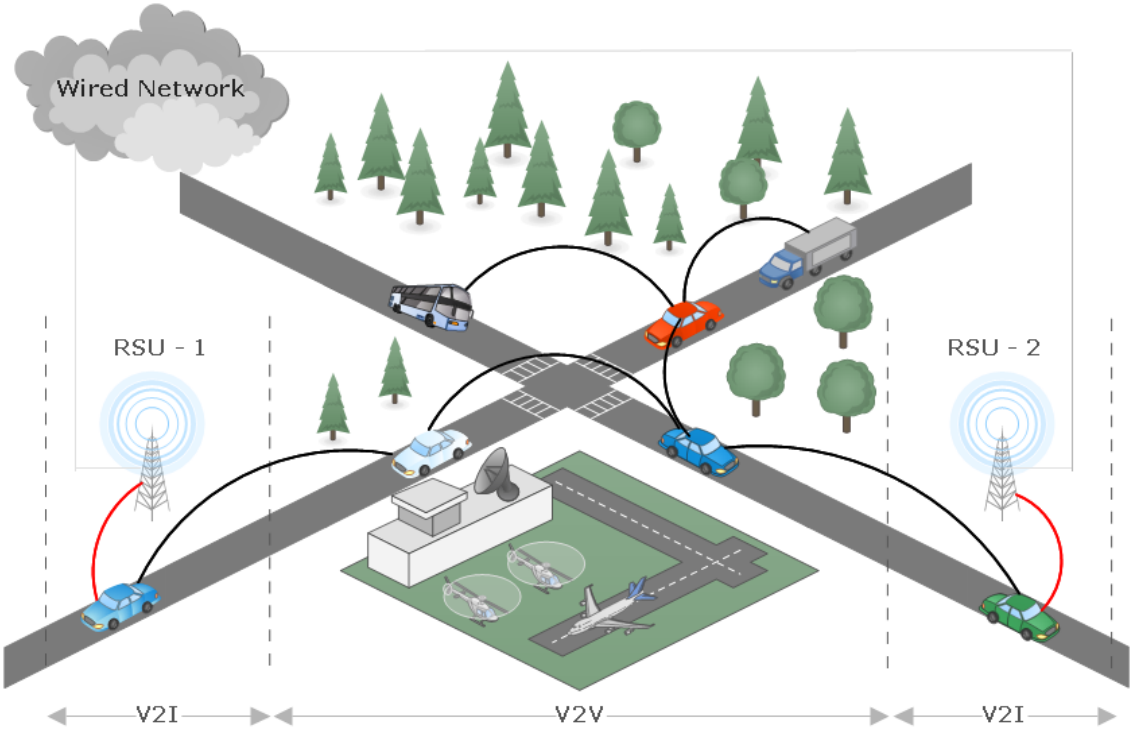

Figure 1.1: A VANET model with RSUs.

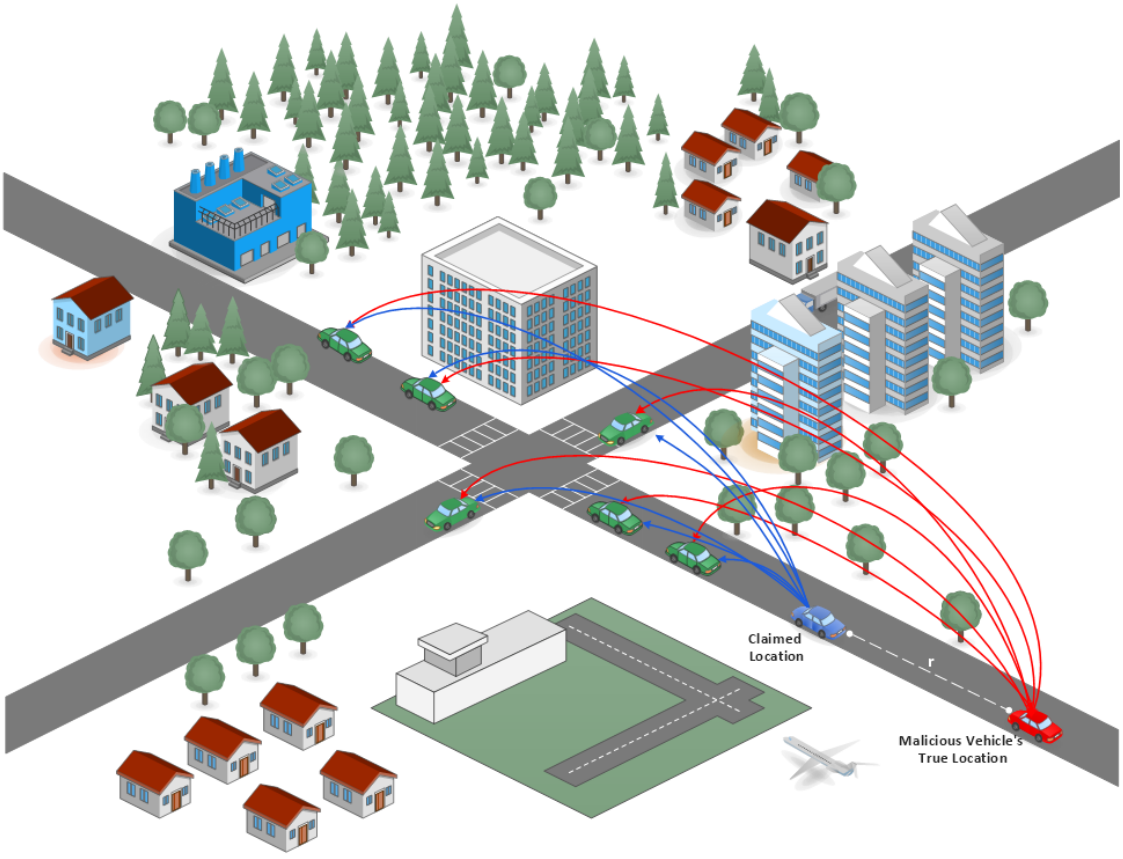

Figure 1.2: A VANET model without RSUs.

avoid traffic collisions [16], and add to the highways' capacity through 'highway platooning'[1]. However, the most important outcome expected from the deployed VANETs is safety. A World Health Organization (WHO) traffic report suggests that the number of global causalities due to road traffic incidents in 2018 alone were 1.35 million [17]. This statistic is 0.11 million higher than the same statistic in 2013 [18]. Globally, road traffic injuries are the 8th leading cause of death for people of all ages and are expected to become the 7th leading cause of death by 2030 [17]. Road traffic crashes are the main cause of death for children and young adults aged 5-29 years [17]. Lowering such a high death toll is a challenge that VANETs are yet to meet.

Location information forms the basis of almost all network decisions in VANETs. For this reason, numerous location verification frameworks have been formulated in the past [1, 2, 19–25].

1.4 LVS *vs.* a Positioning System

There is a significant difference between an LVS and a positioning system. In an LVS, the framework verifies a vehicle's reported position based on some input measurements and issues a binary decision (either legitimate or malicious) [24, 26–28]. This means that the framework will label the vehicle as legitimate if the reported location is considered true, else malicious if the reported location is deemed false. The input measurements refer to the physical layer properties of the vehicle's transmitted signal measured at RSUs, *i.e.*, Time of Arrival (ToA), Received Signal Strength (RSS), and/or Angle of Arrival (AoA). In real-world circumstances, RSS measurements are most desirable due to their ease in collection, robustness, cost-effectiveness, and wider availability. On the other hand, collection of ToA or AoA requires sophisticated equipment. In addition, ToA measurements require clock synchronization at nanoseconds levels. Fig. 1.3 shows a schematic for a general LVS. On the other

[1]Highway platooning is a concept where nearby vehicles on a highway share their steering angle, acceleration speed, and other relevant parameters. Sharing such information helps the vehicles to keep them at a close but minimum safe distance from the surrounding vehicles. With highway platooning, the number of vehicles in a unit area thus increases, increasing the highway's capacity.

hand, the framework outputs an estimated location [29] for the vehicle in a positioning system. Time of flight measurements of the pulsed signals between a transmitter and receiver and a trilateration method is, in general, considered to locate the vehicle within a positioning system. Modern positioning systems can also function following other positioning techniques, *i.e.*, inertial sensing, phase difference, direct field sensing, etc.

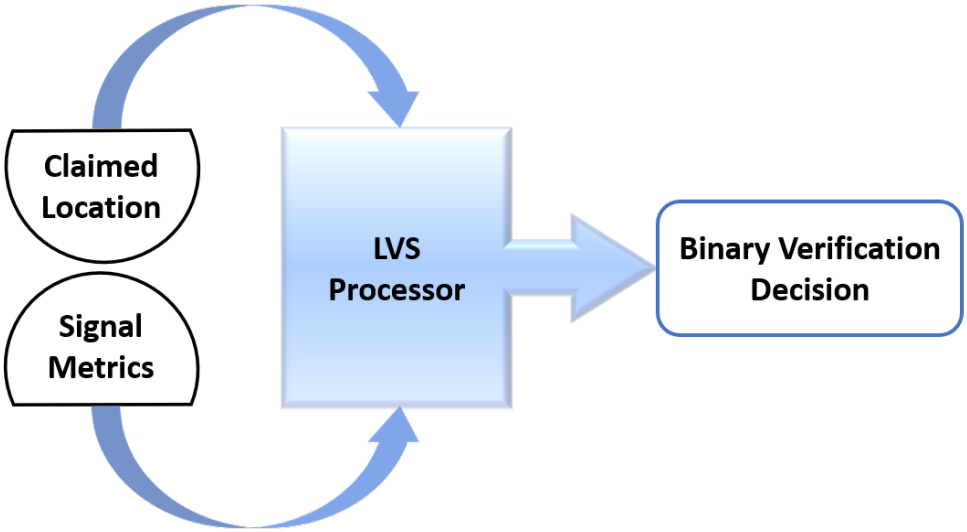

Figure 1.3: A general LVS.

ML has transformed many aspects of human society in recent years. In particular, ML algorithms have shown their strength in two main areas. First, in processing the image data through Convolutional Neural Networks (CNNs), they help with numerous tasks such as facial identification [30], medical diagnosis [31], computer vision [32]. The CNNs are well-known for extracting correlation features in images [33] for onward tasks accomplishment. Second, through processing speech data, ML algorithms help identify a language [34] or a speaker [35]. At most, ML algorithms are utilized to process the speech data in two ways; first, converting the speech data into spectrograms (*i.e.*, images) and then using the CNNs to process these spectrograms [36], second, processing the speech data through Long Short-Time Memory (LSTM) networks [37] towards achieving the desired goal. The LSTM networks are powerful processors of speech data as they are good at extracting hidden features due to the presence of temporal characteristics in such data. To the best of my knowledge, to date, little or no work has been reported where ML algorithms have

been utilized to manipulate physical layer properties (RSS, AoA, ToA, etc.) of the users' or vehicles' transmitted signals. In this thesis, such properties of the vehicles' transmitted signals coupled with various ML techniques are considered to streamline numerous functions of the VANETs. It is anticipated by the research community that ML will play a key role in many aspects of the VANETs, such as vehicle classification problems, vehicle vision systems, autonomous driving, big data analytics, routing, vehicle security, and driver behaviour [38].

1.5 Positioning Algorithms

A Global Navigation Satellite System (GNSS) is used to locate a device/user based on the time of flight in outdoor areas. The GPS (a type of GNSS) measures one way flight time in locating a device/user [29] and therefore requires tighter time synchronization between the clocks of the receiver and satellites. On the other hand, a pinpoint local positioning system [39] measures the Round Trip Time (RTT) of flight during localization, thus eliminating the requirement for time synchronization.

In VANETs, if a vehicle shares its accurate location information with nearby vehicles in an area (and vice versa), it can avoid hidden vehicles and other obstacles even in poor weather conditions [40]. Moreover, the vehicle can detect unsafe lane changes from approaching vehicles with the help of shared information to avoid collisions. During poor visibility conditions, a vehicle entering an intersection may miss another vehicle entering the intersection from a different direction. This miss detection can lead to dangerous and undesirable outcomes. Such undesirable outcomes can be avoided if both the vehicles are aware of each other's position through a prior exchange of location information [41]. In addition, if an accident has already occurred, other vehicles in the area can be informed through a post-accident warning message. Such a message will float the accident location's GPS information, thus helping other approaching vehicles apply emergency brakes to avoid pile-up collisions. However, the floated location information of the accident's site may have GPS errors [42] due to poor weather conditions, poor communications with the satellites

due to surrounding buildings, or some GPS hardware/software issues. Therefore, accuracy in localization in such scenarios is mission-critical.

One solution to improve the accuracy in localization in VANETs is through differential GPS [43]. The distance between the GPS transmitters and receivers is so significant that the signal propagation delay within an area is nearly the same, leading to approximately the same GPS error in receivers. Calculating a differential GPS error in one receiver can help calibrate the nearby GPS receivers. A reference vehicle in the area (with its true location information known) can be utilized to first calculate the GPS error by comparing the data received from the GPS satellites with its known location information. The calculated GPS error can then be used by nearby passing vehicles to fix errors in their positioning information. The differential GPS solution can be helpful in rural areas but may not be practical for deployment in urban areas for two reasons. Firstly, there is a possibility that taller structures in urban areas can result in variations in GPS errors within small blocks and secondly, the high differential GPS deployment costs involved in such areas may not be feasible.

In situations where a vehicle loses its GPS signal temporarily due to poor reception, such as in underground tunnels, indoor parking, or in dense city centers, the vehicle can use the dead reckoning positioning technique [44]. In this technique, the vehicle determines its position based on its last measured GPS location information and other motion parameters. The motion parameters include the vehicle's speed, orientation, steering, and time information, etc. The dead reckoning positioning technique can only help determine the vehicle's location when the vehicle loses contact with the GPS satellites for a short time only.

The authors in [45] developed a radio frequency identification-assisted localization scheme to help vehicles that can face temporary GPS outages. This scheme also assists vehicles with lane-level localization accuracy. The authors utilize localization concepts similar to the differential GPS and help GPS-less vehicles find their location information through contact with their GPS equipped neighbours.

Error in GPS positioning of vehicles due to the Non-Line-of-Sight (NLoS) phenomenon is a serious issue in dense urban areas. This issue can be addressed using

3-D maps, but such maps must be installed and kept updated at the vehicle level. The authors in [46] use shadow maps instead of 3-D maps to tackle this problem. Unlike 3-D maps, shadow maps are available on-demand in VANETs. The Shadow maps efficiently represent the GPS satellites' reception in dense urban areas and assist VANETs with localization to overcome the shortcomings of GPS positioning in such areas. Onfield trials indicate high performance for the proposed approach compared to other techniques.

The authors in [47] propose an online recursive location estimation framework to improve the localization accuracy in VANETs. The proposed framework leverages different radio-ranging distance measurement techniques, V2V communication, and two data fusion methods to achieve real-time localization in VANETs. In the first fusion method, a vehicle's (let's say vehicle-A) belief about its present location is measured using the extended Kalman filter [48]. This belief is further strengthened through communication with the nearby vehicles and by considering the nearby vehicles' beliefs about vehicle-A's location. In the second fusion method, the authors weigh the beliefs of the surrounding vehicles about vehicle-A's location.

The authors in [49] present an experimental onfield proof of a cooperative localization approach to assist GPS-enabled VANETs. The cooperative localization approach works based on V2V communication and ranging measurements. The authors initially validate the accuracy of a one-dimensional V2V ranging on a highway in non-static conditions. Later on, they verify the influence of cooperation on positioning between vehicles. This study is beneficial for VANETs in situations of erroneous GPS initialization and complete GPS denial.

The localization of smaller means of transportation, such as bicycles, will assist future VANETs with their operation. Although a GPS can help locate such small vehicles outdoors, it is challenging to equip them with GPS. This challenge can be due to multiple reasons, such as GPS deployment costs and the low power capabilities of these vehicles. As a way forward, such vehicles can be equipped with sensor nodes, and these sensor nodes can then help with the localization of the vehicles. In addition, the sensor nodes can help with the localization of vehicles in situations

where a GPS cannot function, thus providing VANETs with an extra layer of positioning. A few of the well-known sensor-based localization algorithms that can help locate vehicles in VANETs are discussed next.

The localization technique outlined in [50] can be considered to help position the small vehicles through their attached sensor nodes. To eliminate the costs of the GPS receivers and the power required for operating such receivers, the authors consider small sensor nodes with short-range radio frequency transceivers to assist with the localization. These sensor nodes have limited energy requirements, and their interconnection is taken into account to localize nodes within the network. In addition, a few preset reference sensor nodes with overlapping coverage regions help nodes to self-locate them.

Range-based localization algorithms [51, 52] can help position the small vehicles with the aid of the attached sensor nodes. Although such algorithms are accurate, they are expensive from a deployment perspective as they require specialized hardware. These algorithms take into account the angle information or the absolute point-to-point distance to function. The high costs of the additional hardware make such algorithms particularly impractical in VANETs with a high density of small vehicles. To overcome the high hardware costs (used in the range-based positioning algorithms), range-free localization algorithms [53, 54] are considered cost-effective solutions. Such algorithms are easier to implement, require less hardware, and have low power requirements. The problem with such algorithms, though, is their high positioning error compared to range-based positioning algorithms.

The authors in [51] have formulated a range-based positioning algorithm for sensor nodes that can be utilized in VANETs. An individual mode of sensing in a sensor network may perform poorly due to interference problems from multiple sources in the surrounding environment. To overcome these interference issues, the authors utilize an acoustic ranging system for localization. However, the proposed algorithm results in erroneous readings in NLoS channel situations. The authors then suggest that using evidence from orthogonal sensory channels may help overcome the inaccurate readings in such NLoS channel conditions.

The range-based localization framework proposed in [52] can determine the orientation and location information of the small vehicles in VANETs with the help of the attached sensor nodes based on AoA. The developed framework assumes that a few (not all) sensor nodes need to have localization capabilities. However, all the nodes must measure the AoA based on the communication with their surrounding nodes. The designed framework is scalable due to its localized communication protocol and can also perform satisfactorily in the case of disconnected networks.

The range-free localization algorithm outlined in [53] is inspired by the biological systems that consider chemical gradients to locate cells. The sensor nodes attached to the small vehicles in this algorithm have no prior information on their location or orientation and assign themselves logical coordinates in the context of their global physical location. This localization algorithm assumes a minimum of fifteen neighbouring vehicles with sensor nodes for better performance. The designed algorithm can locate vehicles equipped with sensor nodes reasonably well based on the local nodes' information and local communication. Moreover, the authors of [53] derive that resolution for any coordinate system is determined by its local communication.

The authors in [54] developed a localized, distributed, and hop-by-hop range-free localization algorithm that can supply the location information of all the sensor nodes attached to the vehicles in VANETs in situations where fewer vehicles with sensor nodes can self-locate themselves. This algorithm is particularly helpful in scenarios where a GPS is not functional for various reasons (*i.e.*, power issues, or NLoS conditions etc.). The location information obtained from this algorithm can help avoid extensive routing tables to streamline vehicles' routing in VANETs.

Indoor positioning has gained considerable research interest in recent years due to the flooding of location-based services in indoor environments. Indoor positioning will be vital to streamline localization in VANETs in situations such as indoor parking and traffic management in underground tunnels. WiFi technology, alongside fingerprinting, has been utilized to formulate new algorithms for indoor positioning. However, such algorithms suffer due to high manpower time and costs for offline indoor surveying. Additionally, their inflexibility in tuning to the dynamic changes

in the indoor environments make such algorithms a less attractive choice for deployment. The VANETs can take advantage of the ML-based indoor positioning framework proposed in [55] to overcome these issues. The fast online learning capabilities enable this framework to adapt to the dynamic changes in the surroundings and help in reducing manpower time and costs for offline site surveys.

The ML-based indoor positioning algorithms usually have one hidden layer in them. VANETs can utilize a deep NN-based indoor positioning framework proposed in [56] to leverage the advantages of multiple hidden layers. Following fingerprinting concepts in the first phase, RSS measurements (as part of training data) from numerous vehicles are collected at various BSs offline. Location information of the BSs is added to the training data prior to training the framework. In the online phase, the trained framework can utilize a vehicle's RSS measurements at the BSs to estimate the vehicle's location indoor.

VANETs can benefit from the outdoor ML-based localization algorithm formulated in [57] to enhance localization accuracy and minimize the sensor nodes' power consumption attached to the vehicles. The developed algorithm utilizes the transmission range of the sensor nodes attached to the vehicles in locating the unknown vehicles equipped with sensor nodes in a three-dimensional space. In addition, the algorithm is also able to identify erroneous sensor nodes within the network to enhance localization performance.

1.6 Heuristic Location Verification Algorithms

The 'ECHO' protocol, based on the use of both radio and ultrasound frequencies, has been developed in [58]. The main advantage of this protocol is that it is mainly independent of the tight synchronization (timing) requirements demanded by other schemes. In this protocol, the verifier sends a packet with an unknown random value to the claimant. The claimant instantly responds, via ultrasound and wireless channels, upon receiving the packet. The verifier compares the RTT for both the channels against ideal delay times. If the times-of-flight are less (more) than the

ideal times, the claimant is marked to be inside (outside) the region.

The authors in [59], in an attempt to prevent distance fraud attacks from a user, initially suggest several ways to determine if the user is in the range of a verifier. First, the verifier sends a nonce while controlling the transmission power of the wireless signal. The user can only respond to the nonce if he is within the transmission range of the verifier's transmitted signal. Afterwards, the authors suggest a challenge-response protocol to determine the distance between the user and the verifier. The verifier initially sends a challenge to the user and initiates a timer. The user solves the challenge and sends back his response to the verifier. The verifier stops the earlier activated timer upon receiving a response from the user. Then the verifier multiplies the RTT with the propagation speed of the signal to figure out the verifier-user distance. However, the RTT also includes the challenge processing time *i.e.*, the time it takes the user to solve the challenge. To overcome this issue, the authors suggest either using ultrasound or electromagnetic signals. The adopted tactics help the verifier know if the user is within some range/distance. The verifier, however, cannot find the actual location of the user. To address this limitation, the authors take into account three verifiers. By combining the outcomes of the verifiers and through basic triangulation, the user is located at a point or in a region.

A 'verifiable multilateration' scheme has been proposed in [60]. The scheme is based on the notion of 'distance bounding' (a malicious node can only claim it is further from the verifier, not closer) and makes use of a minimum of three verifiers. The verifiers need not be tightly synchronized. The scheme can be shown to produce verifiable regions of space under a series of different threat models.

In [61], a series of autonomous sensors have been adopted, which look at a range of metrics, such as the anticipated range of communication, the anticipated maximum speed of vehicles, and the anticipated density of vehicles. Using techniques akin to malicious behaviour detection in intrusion detection algorithms, a location verification outcome is derived. The devised location verification protocol is aimed at location-based routing and takes into account neighbour tables. Such tables are

only possible through cooperation between nodes. The location verification protocol devised in [61] works well with the existing infrastructure and does not require extra resources, *i.e.*, additional hardware.

In a geographical routing protocol, a malicious node can falsify its location information. If the falsified location information goes unchecked and is used by other nodes in the area, the very aim of the routing protocol can be compromised. To address this issue, the authors in [62] have devised a location verification protocol. In this protocol, a non-trusted node is not allowed to self estimate its location. Instead, the node generates a localization request to the anchoring nodes in its surroundings. Each anchor node in the vicinity generates a distance estimate to the querying node. Multiple anchor nodes exchange their distance estimates and locate the querying node through triangulation. Finally, this estimated location is shared with the querying node in the form of a certificate which the querying node can exchange with other nodes in the area.

A timestamp embedded within the packet that contains the claimed location has been used to detect malicious vehicles in [63]. In addition, the timestamp check ensures that the received packet is in the appropriate time window. The authors use a rate-limiting mechanism to validate the location of a vehicle. If the rate of the packets sent by the vehicle (whose location is to be verified) exceeds a preset transmission rate, the vehicle is marked as malicious.

In [64], the notion of Covert Base Stations (CBSs) has been introduced into the location verification problem. A CBS is a BS whose location is assumed to be unknown to the attacker. In conjunction with the use of secret keys, the use of CBSs is shown to improve the verification performance significantly. Specifically, the authors have devised a 'position verification protocol' where a Public Base Station (PBS) communicates with a prover node. The PBS initially sends a challenge to the prover node. In response, the prover node sends its position via radio and ultrasound signals. A CBS, present in the area, also receives the signals. The CBS measures the time difference between the times it receives the radio and ultrasound signals and calculates its distance to the prover node. If this distance corresponds to the position

that the prover node has reported, the location of the prover node is validated else vice versa.

In [65] an onboard radar system has been considered to verify a vehicle's claimed location. Taking noise into account, the scheme detailed in [65] separately determines the GPS position tolerance shadow and radar position tolerance shadow - with the algorithm accepting (rejecting) the prover's claimed location if there is (is not) an intersection between the GPS and radar position shadows. The verifier seeks help from its neighbours in situations where its radar cannot directly locate the prover.

The authors in [66] have formulated a Multihop Location Verification Protocol (MHLVP) for NLoS conditions. The MHLVP uses a cooperative multihop approach to verify the locations of vehicles in conditions where direct communication between the vehicles is not available. To elaborate this further, let us assume a node 'A' wants to verify another node 'V', but there is an obstacle between 'A' and 'V'. Node 'A' contacts its first-hop neighbours in its neighbours' list and provides them with the verification request, including the claimed location of the prover node 'V'. Let us assume a node 'C' (in the neighbours' list of 'A') can directly communicate with the node 'V'. Node 'C' first verifies if the requesting node 'A' is in its neighbours' list. Afterwards, node 'C' measures its distance to the node 'V' (d_{CV}) through RSS and compares this distance with the distance calculated using the claimed location of 'V'. If both the distances are comparable, node 'C' will inform node 'A' of the verification. Node 'A' may receive responses from multiple nodes in its neighbours' list. Such multiple responses from the neighbouring nodes increase the location verification confidence of node 'A'.

The authors in [67] have proposed a secure multilateration algorithm for verifying the location claims of nodes in wireless sensor networks. The algorithm uses the 'time of flight' of a signal for its operation. A prover initially submits its claimed location to the verifiers. Each verifier selects a unique transmission frequency to send a challenge bit to the prover. The choice of frequency and challenge bit for each verifier is known to all the other verifiers. In addition, the verifiers also decide the ToA of their challenge bits at the prover's claimed location in advance. All the verifiers

afterwards send their challenge bits in synchronization to the prover's claimed location. If the prover's claimed location is valid, the prover receives all the challenge bits simultaneously and responds to all the verifiers by broadcasting a tone message on the correct frequency. All the verifiers receive and check the prover's response bit to validate the prover's claimed location. The designed algorithm assumes several conditions for its operation, such as; all the verifiers and claimant nodes need to be in the same plane, all the verifiers shall have no synchronization errors, and all the verifiers must have the same processing delay and detection thresholds.

The authors in [68] have formulated an on-spot and an in-region location verification protocol for wireless sensor networks. Both the protocols work based on ordinary sensors and comprise a Verification Centre (VC). The on-spot location verification protocol includes two sub-algorithms: the greedy filtering matrix algorithm and the greedy filtering by trustability-indicator algorithm. These sub-algorithms use matrices to find localization errors in the estimated and claimed locations of sensors. For on-spot location verification, the VC approves a sensor's claimed location if the difference in the sensor's claimed and estimated locations is within an anomaly degree 'D' ('D' is an input to the verification protocol and varies according to the requirement of the application). The VC approves a sensor's claimed location in the in-region location verification protocol if the claimed location is within a 'verification area'. The 'verification area' parameter is supplied to the protocol and can vary between applications.

The authors in [69] have extended the location verification problem to a larger cooperative framework by using other vehicles. The algorithm in [69] utilizes a verifying and a cooperative vehicle to send a challenge message to a prover. The prover responds to the challenge message by sending its claimed location via a radio signal. The verifier can verify the prover's claimed location based on the time of flight of the response signal recorded at the two locations (*i.e.*, at the verifying vehicle and the cooperative vehicle).

In [70], the location of the vehicle has been verified conditioned on a guarantee of privacy to the user. This algorithm uses the beacons transmitted by vehicles in the

area to cross-check the vehicle's claimed location. The devised location verification algorithm can also determine the location of a prover in situations where the prover's claimed location cannot be verified.

The impacts of position cheating attacks on a Fast Multi-hop Broadcast Algorithm (FMBA) [71] have been analyzed in [72]. The FMBA has two phases; an estimation phase and a broadcast phase. The FMBA aims at reducing the travel time of a message from a source to the farthest vehicle. This aim is achieved by cutting down the number of hops a transmitted message needs to travel to the target. However, if vehicles are able to cheat their locations successfully, they can severely influence the performance of the FMBA. The authors in [72], therefore, devise a mechanism to identify attacks from such malicious vehicles. The verifying vehicle takes into account its transmission range and checks if the claimed location of the prover vehicle is within its transmission range. In addition, the verifying vehicle also seeks assistance from collaborative neighbours to authenticate the claimed location of the prover. The devised mechanism does not require any additional hardware.

In [27], a tile-based system has been used to construct aggregated 'belief' type algorithm that includes the use of nearby vehicles as a means to influence the belief value. The algorithm consists of two layers. In layer-1, virtual tiles on the road are considered. These tiles are used for onward verification of the vehicles' claimed locations. A verifying vehicle 'V' calculates the expected RSS from each tile by considering its location and the locations of the tiles. The vehicle 'V' records the RSS measurement from a neighbouring vehicle 'N' and the vehicle N's claimed location. The vehicle 'V' utilizes the claimed location of the vehicle 'N' to approximate vehicle N's presence on virtual tiles. The vehicle 'N' passes the layer-1 verification if the measured and calculated RSS meats a probability threshold (that is set for the vehicle N's presence on a virtual tile). After clearing the layer-1 verification, the vehicle 'N' is subject to a second layer verification. In layer-2, transferable belief model-based verification is performed, which is related to the concept of verification of a vehicle's presence in the neighbourhood of other neighbouring vehicles. These layers are collectively considered to verify the vehicle's claimed location.

To reduce the impact of erroneous location information on future location predictions for smooth geographical routing, the authors in [73] have proposed a protocol named Location Error Resilient Geographical Routing (LER-GR). The protocol of [73] utilizes the errors in location information on nearby vehicles in choosing the next forwarding vehicle to improve the geographical routing and overall network performance.

The use of token-based registration models and multiple (and independent) authentication rounds has been discussed in [74]. The authors have termed their on-spot location verification scheme for wireless sensors/nodes as Mutually Shared Region-based Location Verification (MSRLV) system. MSRLV is a two-step process. In the first step, a verifier carries out an integrity check for the claimant nodes through a Distance-inconsistency filtering (D-filtering) procedure. In D-filtering, a verifier finds the difference between measured and estimated distances for claimant nodes. The claimant nodes that qualify the D-filtering phase are not entirely trusted and are subject to a second step. In the second step, the verifier and the claimant nodes utilize respective Mutually Shared Region (MSR) tokens to authenticate the locations of the claimant nodes. An asymmetrical MSR token is calculated independently by the verifier V and claimant M based on the random values attached to the recently reported data packets from their common neighbours in the MSR region.

Power-optimization and network overhead issues, in the context of location verification, have been examined in [75]. The authors have introduced an activation policy that only triggers specific nodes for verifying a prover's location. In the process of an attack, an attacking node transmitting from location 'A' claims to be at location 'B'. The verification protocol considers a binary hypothesis model and utilizes the Likelihood Ratio Test (LRT) to prove if the attacking node's claim is genuine or otherwise.

The authors in [76] have used vehicle trajectories to aid to the location verification of vehicles in vehicular networks.

The authors in [77] have devised a location verification protocol to streamline the exchange of secure messages among vehicles. The protocol considers vehicles

in clusters on a highway. A central authentication unit first validates a cluster head. Once validated, the cluster head verifies all the other vehicles in the cluster. Thus, the exchange of messages only happens once all the vehicles in the cluster are authenticated.

The authors in [78] have designed a location verification algorithm to detect malicious nodes for a range-based localization in wireless sensor networks. The algorithm works on the basis of RSS and ToA of the nodes' transmitted signals and comprises four stages. In the first stage, location information data of the nodes is collected through trilateration. In the second stage, the collected data is divided into normal and abnormal clusters. In the third stage, two reference distances between the unknown node (whose location information is unknown) and its neighbouring nodes are obtained through RSS, and ToA-based distance measuring approaches. An indication of the presence of malicious nodes is triggered if the difference between the two measured distances exceeds a preset acceptance threshold. All the information from the malicious nodes is discarded in the fourth stage, leaving behind information from legitimate nodes only. The unknown node is afterwards located through multilateration on the distance information from the legitimate nodes. The authors have extended the performance of their designed algorithm through another location verification framework. The new framework improves the detection performance of the previously discussed algorithm.

1.7 Information-Theoretic LVSs

The algorithms discussed thus far set their own performance goals and did not identify optimal performance thresholds for LVSs in any context. Therefore, a few notable works where considerable efforts are made in designing optimal LVSs are highlighted next. The design of such LVSs is based on information-theoretic constructs.

The authors in [79] have designed an optimal information-theoretic LVS framework to verify vehicles' claimed locations in VANETs. The authors aim to maximize

the mutual information between inputs and outputs of the LVS. The LVS considers a vehicle's claimed location and the independent RSS measurements of the vehicle's transmitted signals made at numerous BSs as inputs. The authors formulate an information-theoretic decision rule to validate the vehicle's claimed location. Towards the end, the performance of the LVS is studied and validated with the devised decision rule in situations where a malicious vehicle does (does not) optimize its transmit power.

The authors in [80] capitalize on the work in [79] and have introduced the use of directional antennas to enhance the LVS performance. The authors take into account a dual-slope large scale fading propagation model for their work. The authors claim this propagation model to be more suitable for VANETs, in general. The authors consider as input the information on which a BS measures the largest RSS from a vehicle, in addition to the vehicle's claimed location and the RSS measurements at multiple BSs. This work assumes that a malicious vehicle is unable to optimize its parameters.

The authors in [26] have extended their earlier work in [80]. The system model now assumes a malicious vehicle has extended capabilities to optimize its RSS measurements at the BSs. The authors derive close form expressions for the detection capabilities of the LVS for a few scenarios and report an equal performance for the analytical and simulation-based extracted results.

The authors in [24] have designed and studied the performance of an information-theoretic RSS-based LVS under a log-normal propagation model. It is shown that this LVS significantly improves its performance due to the shadowing correlation between the RSS measurements made at two different locations. The authors consider RSS and Differential Received Signal Strength (DRSS) measurements for their study. They take into account a binary hypothesis model and the LRT for their analysis. They report an identical performance for the RSS and the DRSS-based LVSs under correlated shadowing. Additionally, the performance for both the LVSs remains the same when a malicious vehicle does not optimize its attack location. However, the RSS-based LVS outperforms the DRSS-based LVS when the malicious

vehicle cannot optimize its transmission power and other controllable parameters.

The authors in [81] have formulated a general LVS scheme where a single BS equipped with multiple antennas identifies a malicious vehicle in VANETs. The authors carry out a detailed analysis of location verification under realistic settings of Rician fading channels. It is shown that the detection performance of the LVS is independent of the channel conditions between the BS and the malicious vehicle, provided the malicious vehicle's antennas count exceeds a specific limit.

In [82], a Location Spoofing Detection System (LSDS) has been formulated to verify a vehicle's claimed location. The LSDS's performance is examined based on the impact of the channel's Line-of-Sight (LoS) component between a BS and vehicles. The LSDS shows better performance as the number of antennas on the BS or genuine vehicles increases. The system also performs efficiently as the Rician K-factor[2] of the channel between the BS and genuine vehicles enhances. However, a sensitive observation is reported towards the end, which says that the LSDS detection capabilities decrease when the number of antennas on a malicious vehicle gets equivalent to that of a genuine vehicle.

The authors in [83] have considered several information-theoretic approaches such as Renyi divergence, Renyi mutual information, Kullback-Leibler (KL) divergence, KL mutual information and Jensen-Shannon divergence to formulate an optimum decision threshold for an LVS. An LVS that follows the newly identified threshold derived based on Jensen-Shannon divergence can enhance the overall performance through better detection rates.

1.8 Why Machine Learning?

A standard limitation exists in all the discussed LVSs [23, 24, 79, 81]. In order to operate, they require the environment or channel conditions assumed at the time of their design. For instance, a number of traditional LVSs require information on

[2]A Rician K-factor is a ratio of the power of LoS component to the power of scattered components of the channel.

numerous channel parameters in advance for their functioning. This includes an *a-priori* knowledge on

1. the proportions of legitimate and malicious users in the field,
2. the path loss exponent of the channel between users and RSUs,
3. the parameters describing the distribution of noise (*i.e.*, standard deviation, etc.) affecting the RSS, ToA, and AoA measurements (of the users' transmitted signals) at different RSUs,
   etc.

Such assumptions are far from reality in real-world situations. Moreover, a few of the discussed algorithms address specific types of attacks only [21]. All these factors limit their operational capabilities in practical scenarios. More recently, ML algorithms have been considered to validate the location of users in a region [84–86]. No author has attempted to consider ML algorithms to precisely verify the reported locations of users, nodes, or vehicles, etc., at their claimed locations. This thesis closes this gap by integrating ML algorithms to design intelligent LVS solutions for precision location verification. Specifically, NNs are used to formulate intelligent LVS frameworks, and such frameworks are proven to work through simulated as well as experimental data.

1.8.1 Artificial Intelligence, Machine Learning, and Neural Networks

Artificial Intelligence (AI) is a broader term and refers to any task where a computer mimics human behaviour to accomplish the task intelligently. One example is a computer playing a chess game with a human. AI itself does not tell what techniques it uses to play the chess game against a human. These techniques can be rule-based or hard-coded, etc. ML is also one such technique that let a computer establish some task through human-like behaviour.

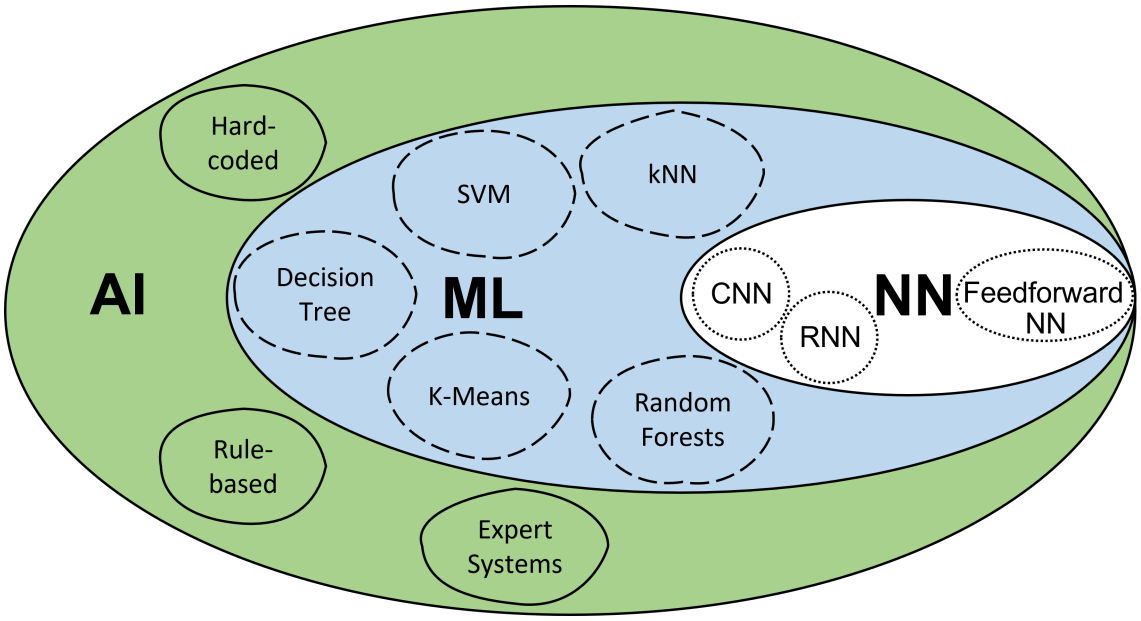

Figure 1.4: A figure showing the relationship between AI, ML, and NNs.

The researchers in the early days of AI found a few tasks (*i.e.*, image recognition, extracting meaning from text, etc.) too complex to be handled with fixed rule-based, hard-coded, and other such algorithms. At this point, the researchers concluded that computers not just mimic human behaviour but mimic how humans learn. This idea allowed the researchers to think of ML (algorithms). One example of ML is supplying a large amount of measurement data of vehicles' signals to a computer, with some vehicles marked as malicious and others as legitimate. The computer is tasked to learn the type of vehicles from the supplied data. This learning exercise helps the computer classify a vehicle as legitimate or malicious when supplied with new unseen data.

However, it was not easy for the simple ML algorithms to tackle some of the problems humans found easy. Recognizing handwriting or a person's speech were examples of a few such problems. The researchers thought if ML was all about training computers to learn like humans, why not go all the way to let computers mimic the human brain? This resulted in the concept of NN algorithms. NN algorithms produced promising results for complex problems which ordinary ML algorithms failed to address. The relationship between AI, ML, and NNs is shown is Fig. 1.4.

The majority of chapters in this thesis are related to the classification of vehicles

for which ML algorithms other than NNs, *i.e.*, Support Vector Machines (SVMs), etc., can be considered. The use of NNs is preferred over all the other ML algorithms. Let us compare NNs with SVMs. Below is a list of a few advantages that the NNs have over the SVMs

- NNs can handle a multi-class classification problem by assigning probabilities to each class. In contrast, an SVM model can address such a problem through one-*vs.*-all classifiers. For example, if a classification problem has fifteen categories to classify, a single NN can be used for the classification task. On the other hand, we will need fifteen SVM models to address the same problem.
- In an NN model, the inputs, the number of neurons in the hidden layer(s), and the outputs are fixed. In an SVM model, the number of support vector lines may reach the count of instances.
- Unlike NNs, an SVM model may not perform well when the number of input features is greater than the number of examples in the training dataset.
- An NN is capable of extracting hidden features from the input data. On the other hand, one has to manually identify all the features in the input data for an SVM model to perform adequately.

The SVM models offer a few advantages over NNs, but overall, NNs provide more flexibility and benefits than SVM and other ML classification-focused models. As such, NNs will be considered in the chapters ahead.

As NNs will be discussed frequently in this thesis; the basics of NNs are summarized next.

1.8.2 A Neural Network

An NN is a branch of ML that is inspired by the functionality of the human brain. An NN is formed through a combination of intermediary units known as neurons or perceptrons. A neuron in an NN roughly mimics the functionality of a neuron in the

human brain. A neuron in the human brain is triggered to issue an instruction or aid to a combined instruction as soon as it is activated through an input signal. Similarly, to achieve a combined goal for the NN, a neuron in the NN transmits a signal to the connected neurons upon receiving an input signal. In the case of the NN, the input signal to the neuron is a real number. Neurons have thresholds and can activate a signal at its output once the threshold is reached. All the neurons collectively help the NN identify patterns and learn complex relationships between the inputs and outputs in the supplied data. Identifying patterns and learning complex relationships between the inputs and outputs is usually termed as the training phase of the NN. The NN, after training, can manipulate previously unseen and unknown data in the testing phase.

There are numerous branches of NNs such as Feedforward neural networks, CNNs, Recurrent Neural Networks (RNNs), etc. Since this thesis considers feedforward NNs for the studies carried out in the chapters to follow, the discussion ahead is mainly in the context of a feedforward NN. A feedforward NN is a type of artificial NN where connections between the neurons do not form a cycle. Moreover, the input information flows in the forward direction only. A feedforward NN has an input layer, an output layer with one or more outputs, and a single (or multiple) hidden layer(s) with one or more neurons. The activation in the output layer determines how much the system thinks that a given input corresponds to a given output. The number of neurons in the hidden layers and the number of hidden layers themselves is an arbitrary choice related to the accuracy, robustness and fitting of the system and can be best chosen after executing the same experiment with different network architectures [87]. One idea is that some but not all hidden features in the input data are recognized by the first hidden layer, which helps the next hidden layer identify more features until an output at the output layer is achieved, something that is close to the ground truth.

Let us assume an NN that has two hidden layers and a single output in the output layer. Considering this NN architecture, activation in neurons in the input layer determines the activation in neurons in the first hidden layer, activation in neurons in the first hidden layer determines activation in neurons in the second

hidden layer. Finally, activations in neurons in the second hidden layer determine the activation of the only neuron in the output layer towards producing a single output value. This can be analogous to the biological theory that some neurons firing causes other to fire. The pattern of activations in the input layer causes some precise pattern identification in data in the first hidden layer, which onwards causes further pattern recognition in data in the second hidden layer. Finally, based on the activations from the second hidden layer, an output value is derived at the output layer.

The output of a neuron in a layer acts as an input to the neurons in the layer next to it. In order to determine an appropriate strength for these outputs from the input layer to the first hidden layer, from the first hidden layer to the second hidden layer and from the second hidden layer to the output layer, certain parameters are used, which are usually referred to as 'weights'. Weights are just numbers that can be positive or negative. We can think of these weights as tuning knobs whose value gets set to optimum values with the training of the NN. The weights from one layer determine the activation in the next layer until an output is predicted at the output layer. Optimum values of the weights in the intermediate layers would result in an output close to the ground truth. Assuming the number of neurons in the first and second layers is the same (*i.e.*, N), an activation for the n_{th} neuron in the second layer is derived as

$$a_{2n} = w_{11}a_{11} + w_{12}a_{12} + \cdots + w_{1N}a_{1N}, \quad (n = 1,\ 2,\ 3, \ldots, N), \tag{1.1}$$

where a_{2n} is the activation of n^{th} neuron in the second layer, w_{11} is the weight connecting a_{11} (1^{st} neuron in the first layer) to the n^{th} neuron in the second layer, w_{12} is the weight connecting a_{12} (2^{nd} neuron in the first layer) to the n^{th} neuron in the second layer, and w_{1n} is the weight connecting a_{1n} (n^{th} neuron in the first layer) to the n^{th} neuron in the second layer. The activation a_{2n} because of the above equation can be any number from negative infinity to positive infinity. A nonlinear function is utilized to map this activation value to a smaller scale, *i.e.*, from 0 to 1 or from -1 to 1, etc. This nonlinear function is commonly known as a transfer function or an activation function. A few well-known transfer functions

are the logistic sigmoid transfer function, the tangent sigmoid transfer function, and the Rectified Linear Unit (ReLU) transfer function. The logistic sigmoid transfer function scales an input value x to a value between 0 and 1 at the output, *i.e.*, a too negative x is mapped close to 0, and a too positive x is mapped close to 1. The mapped values steadily increase around the input value equivalent to 0. Similarly, the tangent sigmoid transfer function scales x to a value between -1 and 1 at the output, and the ReLU transfer function scales x to 0 (if $x < 0$) or x (if $x > 0$) at the output. Mathematically, a logistic sigmoid transfer function for a given input x, $\phi(x)$, is defined as

$$\phi(x) = \frac{1}{1 + e^{-x}}. \tag{1.2}$$

A tangent sigmoid transfer function for a given input x, $\psi(x)$, is defined as

$$\psi(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}. \tag{1.3}$$

And a ReLU transfer function for a given input x, $\sigma(x)$, is defined as

$$\sigma(x) = max(0, x). \tag{1.4}$$

A graphical representation of the transfer functions discussed above is given in Fig. 1.5

An NN can be used for two types of learning; unsupervised learning or supervised learning. The main difference between these two types of learnings is that in unsupervised learning, each set of features representing an example within the training dataset has no ground truth label or target included. One example of unsupervised learning is detecting defective mechanical parts in a factory. In contrast, in supervised learning, each set of features representing an example in the training dataset has a ground truth label or target included. What is a label or a target? We can think of a dataset of images of dogs and cats. To supply this dataset as input to train an NN for object recognition, we need to specify to the NN the object in each image, *i.e.*, if an image contains a dog, the label (target) for that image is 'a dog'. Similarly, if an image has a cat, the label (target) for that image is 'a cat'. We can represent labels (targets) through numerical values. For example, a numerical zero

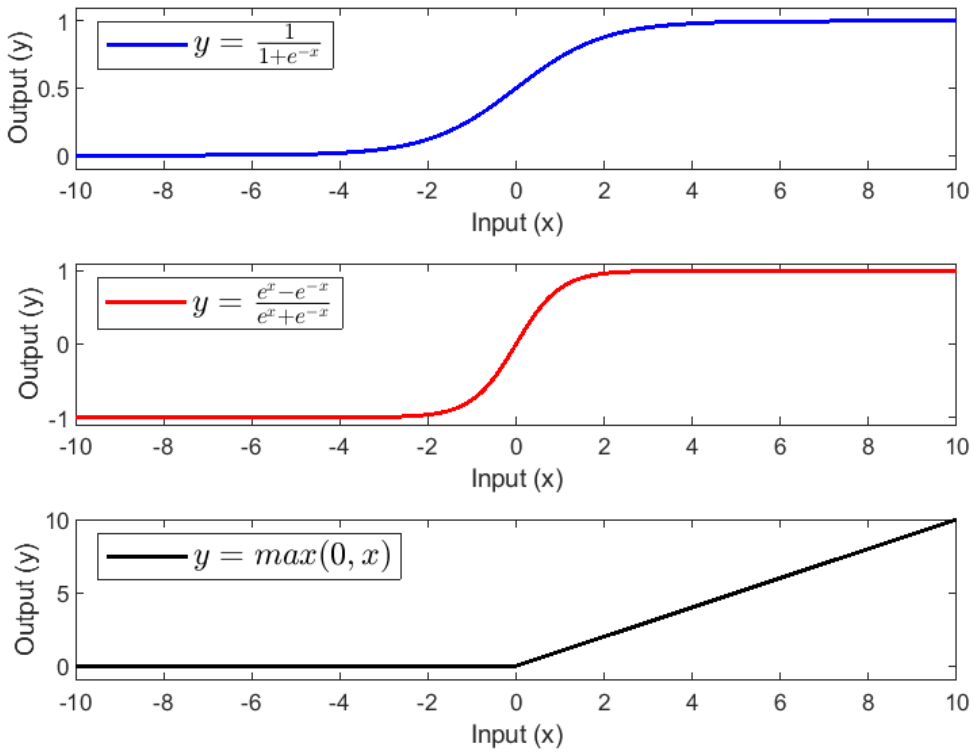

Figure 1.5: Graphical representation of the output (y) of various transfer functions for a given input (x): a logistic sigmoid transfer function **(Top plot)**, a tangent sigmoid transfer function **(Middle plot)**, a ReLU transfer function **(Bottom plot)**.

(0) can represent 'a dog', and a numerical one (1) can represent 'a cat'.

1.8.3 Neural Networks for Supervised Learning

This section briefly highlight NNs used for supervised learning. Supervised learning NNs can help in different types of analysis. Two main areas where a supervised learning-based NN helps are; regression and classification.

1.8.3.1 Regression

We can think of an NN to learn a function f such that $y = f(\mathbf{x})$, where $\mathbf{x} = x_1, x_2, x_3, \ldots, x_m$, is the supplied input vector with m features. In regression analysis, the output y is a real number, *i.e.*, $y \in \mathbb{R}$. An example of an NN performing a

regression task is determining a vehicle's location information (*i.e.*, x and y coordinates) in VANETs. To explain the regression concepts further, a linear regression is discussed to find the location coordinates of a vehicle. For this purpose, an NN is used to predict the location coordinates (x and y) of a vehicle in a 2-D plan. For the ease of understanding, the NN used for regression analysis is assumed to predict the x-coordinate of the vehicle in this phase only. The steps for the NN to predict the y-coordinate are exactly the same. The NN takes the signal metrics of the vehicles transmitted signals, $\mathbf{x}$, measured at multiple BSs as input and outputs a scaler $\hat{y}$ (the predicted x-coordinate of the vehicle). Mathematically

$$\hat{y} = \mathbf{W}^T \mathbf{x}, \tag{1.5}$$

where $\mathbf{W} = [w_1,\ w_2,\ w_3, \ldots, w_m] \in \mathbb{R}^n$ are the associated weights that determine how each input affects the prediction $\hat{y}$. For the i_{th} input x_i, if the associated weight w_i has a more positive value, x_i will influence $\hat{y}$ more. Similarly, a more negative value for w_i will decrease the influence of x_i on $\hat{y}$. If $w_i = 0$, the input x_i will not contribute to the overall prediction.

A training dataset is used to train an NN. The training dataset has the ground truth labels (*i.e.*, the vehicle true location coordinates) for all k vehicles. In the training phase, the NN knows the output (*i.e.*, x-coordinate of the vehicle's location) for the supplied input features (representing a k_{th} vehicle). The NN, therefore, tries to minimize the difference between the true x-coordinate and the predicted x-coordinate (that it predicts using its internal parameters) of the vehicle. One way the NN can minimize this difference is through minimizing the Euclidean distance (d) between the NN's predicted and the true x-coordinates as below

$$d_{tr} = \frac{1}{k} ||(\hat{\mathbf{y}_{tr}} - \mathbf{y}_{tr})||_2^2, \tag{1.6}$$

where d_{tr} is the training dataset error, $\hat{\mathbf{y}_{tr}}$, and $\mathbf{y}_{tr}$ are vectors with the predicted, and true x-coordinates for all k vehicles in the training dataset, respectively. The goal for the NN is to minimize d_{tr} to zero, which is only possible when $\hat{\mathbf{y}_{tr}} = \mathbf{y}_{tr}$. Once trained, the NN can be subject to evaluation through a test dataset. Unlike

the training phase, the x-coordinates are not visible to the NN in the testing phase. Through testing, we know how well our trained NN performs. The target for the NN in the training phase is to minimize d_{tr} to zero, which can be achieved by tuning $\mathbf{W}$. In other words, $d_{tr} \to 0$ when d_{tr}'s gradient for the training set with respect to $\mathbf{W}$ becomes zero [88]. That is

$$\nabla_{\mathbf{w}} d_{tr} = 0. \tag{1.7}$$

We can find optimum $\mathbf{W}$ through solving Eq. (1.7) as below

$$\nabla_{\mathbf{W}} \frac{1}{k} ||(\hat{\mathbf{y}_{tr}} - \mathbf{y}_{tr})||_2^2 = 0, \tag{1.8}$$

$$\frac{1}{k} \nabla_{\mathbf{W}} ||(\mathbf{x}_{tr}\mathbf{W} - \mathbf{y}_{tr})||_2^2 = 0, \tag{1.9}$$

$$\nabla_{\mathbf{W}} (\mathbf{x}_{tr}\mathbf{W} - \mathbf{y}_{tr})^\top (\mathbf{x}_{tr}\mathbf{W} - \mathbf{y}_{tr}) = 0, \tag{1.10}$$

where $\mathbf{x}_{tr}$ represents the supplied training dataset input vector.

$$\nabla_{\mathbf{W}} (\mathbf{W}^\top \mathbf{x}_{tr}^\top \mathbf{x}_{tr} \mathbf{W} - 2\,\mathbf{W}^\top \mathbf{x}_{tr}^\top \mathbf{y}_{tr} + \mathbf{y}_{tr}^\top \mathbf{y}_{tr}) = 0, \tag{1.11}$$

$$2\,\mathbf{x}_{tr}^\top \mathbf{x}_{tr} \mathbf{W} - 2\,\mathbf{x}_{tr}^\top \mathbf{y}_{tr} = 0, \tag{1.12}$$

$$2\,\mathbf{x}_{tr}^\top \mathbf{x}_{tr} \mathbf{W} = 2\,\mathbf{x}_{tr}^\top \mathbf{y}_{tr}, \tag{1.13}$$

$$\mathbf{W} = (\mathbf{x}_{tr}^\top \mathbf{x}_{tr})^{-1} \mathbf{x}_{tr}^\top \mathbf{y}_{tr}. \tag{1.14}$$

A bias term, b, can be added to Eq.(1.5) as below

$$\hat{y} = \mathbf{W}^T \mathbf{x} + b. \tag{1.15}$$

This additional term makes the mapping from the input features to the output's prediction an affine function[3]. The bias is used to adjust the weighted sum of the input features to a neuron and helps in either advancing or delaying the neuron's activation. The number of outputs for the above NN can be increased to two, *i.e.*, predict both x and y coordinates simultaneously.

[3]An affine function in the context of a linear model means that the prediction for the model is still a line, but the line doesn't need to pass through the origin [88].

One aspect of the training of an NN model is to minimize the error on the training dataset, but equally crucial for the model is to perform well on previously unseen inputs, *i.e.*, a test dataset. The ability of the NN model to perform efficiently on new unseen data is known as a generalization, *i.e.*, $d_{tr} \approx d_{test}$ (d_{test} represents the NN's test dataset error). This leads to two important concepts; underfitting and overfitting. Underfitting occurs when the NN model cannot obtain an adequate error over the training dataset. Overfitting happens when the gap between d_{tr} and d_{test} is too large. A reasonably trained NN aims to avoid underfitting, overfitting and to generalize well on test datasets.

1.8.3.2 Classification

In classification analysis, an NN identifies which of the Y categories a given input vector $\mathbf{x}$ relates to, *i.e.*, $y = f(\mathbf{x})$. The variable y here is a category represented by a numeric code that belongs to one of the Y categories, *i.e.*, $y = 1,\ 2,\ 3, \ldots, Y$, and f represents the classification NN function. An example (which we will consider ahead in explaining the concepts related to NN used for classification) is classifying a vehicle as either legitimate or malicious through given input data in VANETs. In both the above examples of a classification NN, Y is equal to two, *i.e.*, two categories. A classification NN assigns a probability to each category within Y based on the supplied inputs. The NN outputs the category assigned with the largest probability of all the Y categories. The performance of a classification NN can be measured through 'Total Error'. A Total Error of a classification NN is the ratio of the number of incorrect classifications (false positives and false negatives) to the total number of classifications performed.

1.9 Working of a Feedforward Neural Network

This section provides a brief explanation for the internal working of a single hidden layer feedforward NN. This NN is used for binary classification of a vehicle as either legitimate or malicious in the context of VANETs. The NN has m inputs and

n hidden layer neurons. The inputs comprise the measurements made by RSUs around the physical layer properties of the vehicles transmitted signals, and the vehicles claimed locations, etc. This NN uses a tangent sigmoid transfer function in the hidden layer and a logistic sigmoid transfer function in the output layer.

1.9.1 Forward Pass

In the forward pass, the weighted sum of the inputs for each neuron in the hidden layer is calculated. The weights are initiated randomly. Let $z_1^{[1]}$ represents the weighted sum of the inputs $\mathbf{x}$ (where $\mathbf{x} = [x_1,\ x_2,\ x_3, \ldots, x_m]^\top$) for the first neuron in the hidden layer, then

$$z_1^{[1]} = w_{11}^{[1]}x_1 \ + \ w_{12}^{[1]}x_2 \ + \ w_{13}^{[1]}x_3 \ + \ldots, + \ w_{1m}^{[1]}x_m \ + \ b_1^{[1]}, \tag{1.16}$$

$$z_1^{[1]} = \mathbf{w}_1^{[1]}\mathbf{x} \ + \ b_1^{[1]}, \tag{1.17}$$

where $\mathbf{w}_1^{[1]} = [w_{11}^{[1]} + w_{12}^{[1]} + w_{13}^{[1]}+, \ldots, +w_{1m}^{[1]}]$, and $b_1^{[1]}$ is a bias term added to the weighted sum of the inputs. The weighted sum can be larger in magnitude but may not result in the neuron's activation. Moreover, the neuron may only be activated when the weighted sum is larger or smaller by some constant value. Such a constant value is referred to as bias. The bias term determines how high or low a weighted sum needs to be before the individual neuron gets meaningfully active. Similarly, the weighted sum of the inputs for the second neuron in the hidden layer is $z_2^{[1]} = \mathbf{w}_2^{[1]}\mathbf{x} + b_2^{[1]}$, for the third neuron in the hidden layer is $z_3^{[1]} = \mathbf{w}_3^{[1]}\mathbf{x} + b_3^{[1]}$, and for the n_{th} neuron in the hidden layer is $z_n^{[1]} = \mathbf{w}_n^{[1]}\mathbf{x} + b_n^{[1]}$. The weighted sum of all neurons in the hidden layer are combined as below

$$\mathbf{z}^{[1]} = \mathbf{W}^{[1]}\mathbf{x} \ + \ \mathbf{b}^{[1]}, \tag{1.18}$$

where $\mathbf{z}^{[1]} = [z_1^{[1]}, z_2^{[1]}, z_3^{[1]}, \ldots, z_n^{[1]}]^\top$, $\mathbf{W}^{[1]} = [\mathbf{w}_1^{[1]}, \mathbf{w}_2^{[1]}, \mathbf{w}_3^{[1]}, \ldots, \mathbf{w}_n^{[1]}]^\top$, and $\mathbf{b}^{[1]} = [b_1^{[1]}, b_1^{[1]}, b_1^{[1]}, \ldots, b_n^{[1]}]^\top$. Next, $\mathbf{z}^{[1]}$ is passed through the tangent sigmoid transfer function. Let $a_n^{[1]}$ be the output of the tangent sigmoid transfer function for the n_{th}

neuron in the hidden layer, then

$$a_n^{[1]} = \psi(z_n^{[1]}) = \frac{e^{z_n^{[1]}} - e^{-z_n^{[1]}}}{e^{z_n^{[1]}} + e^{-z_n^{[1]}}}. \tag{1.19}$$

The outputs of all the hidden layer neurons after the application of the tangent sigmoid transfer function are combined as $\mathbf{a}^{[1]} = [a_1^{[1]}, a_2^{[1]}, a_3^{[1]}, \ldots, a_n^{[1]}]^\top$.

Following similar steps as above, the weighted sum, $\mathbf{z}^{[2]}$, for the output layer is calculated as

$$\mathbf{z}^{[2]} = \mathbf{W}^{[2]}\mathbf{a}^{[1]} \; + \; \mathbf{b}^{[2]}, \tag{1.20}$$

where $\mathbf{z}^{[2]} = [z_1^{[2]}, z_2^{[2]}, z_3^{[2]}, \ldots, z_o^{[2]}]^\top$, $\mathbf{W}^{[2]} = [\mathbf{w}_1^{[2]}, \mathbf{w}_2^{[2]}, \mathbf{w}_3^{[2]}, \ldots, \mathbf{w}_o^{[2]}]^\top$, and $\mathbf{b}^{[2]} = [b_1^{[2]}, b_1^{[2]}, b_1^{[2]}, \ldots, b_o^{[2]}]^\top$. The variable o represents the number of outputs. The dimensions of $\mathbf{x}$, $\mathbf{z}^{[1]}$, $\mathbf{W}^{[1]}$, $\mathbf{b}^{[1]}$, $\mathbf{a}^{[1]}$, $\mathbf{z}^{[2]}$, $\mathbf{W}^{[2]}$, and $\mathbf{b}^{[2]}$ are $m \times 1$, $n \times 1$, $n \times m$, $n \times 1$, $n \times 1$, $o \times 1$, $o \times n$, and $o \times 1$, respectively. The number of weights in this NN is $(m * n) + (n * o)$. Similarly, the number of biases for the NN is $m + n + o$. Since this NN here is used for binary classification of a given vehicle (*i.e.*, either legitimate or malicious), the second layer (also the output layer) comprises of a single neuron, *i.e.*, $o = 1$. This makes $\mathbf{z}^{[2]}$ and $\mathbf{b}^{[2]}$ scalar quantities, *i.e.*, $z^{[2]}$ and $b^{[2]}$. The output of the second layer (the output layer) after the logistic sigmoid transfer function is also a scalar quantity, *i.e.*, $a^{[2]}$, and is given as

$$a^{[2]} = \phi(z^{[2]}) = \frac{1}{1 + e^{-z^{[2]}}}. \tag{1.21}$$

The dimensions of $a^{[2]}$ are $o \times 1$. Let $\hat{y}$ denote the predicted output of the NN model, then $\hat{y} = a^{[2]}$. In summary, the sequence of calculations in a forward pass for the NN is $\mathbf{x} \rightarrow \mathbf{z}^{[1]} \rightarrow \mathbf{a}^{[1]} \rightarrow \mathbf{z}^{[2]} \rightarrow \mathbf{a}^{[2]}$.

1.9.2 Backward Pass

The goal of the NN, during the training phase, is to make the predicted output for the i_{th} example ($\hat{y}_i$) in the training dataset as close to the ground truth value (y_i) as possible, *i.e.*, $\hat{y}_i \approx y_i$. However, this is only possible when optimum parameter values

for $\mathbf{W}^{[1]}$, $\mathbf{b}^{[1]}$, $\mathbf{W}^{[2]}$, and $\mathbf{b}^{[2]}$ are taken into account. A backpropagation algorithm is used to derive reasonable parameter values for $\mathbf{W}^{[1]}$, $\mathbf{b}^{[1]}$, $\mathbf{W}^{[2]}$, and $\mathbf{b}^{[2]}$.

A backpropagation algorithm as part of the training phase of the NN is a backward pass algorithm to calculate the gradients of a loss function, $\mathcal{L}$, with respect to $\mathbf{W}^{[1]}$, $\mathbf{b}^{[1]}$, $\mathbf{W}^{[2]}$, and $\mathbf{b}^{[2]}$. A logistic loss function is considered for the NN here. This loss function is, in general, recommended for a classification problem like the one under consideration here. The logistic loss function is given as

$$\mathcal{L}_i = -\Big(y_i \log\left(\hat{y}_i\right) + (1 - y_i) \log\left(1 - \hat{y}_i\right)\Big). \tag{1.22}$$

The above equation can also be written as

$$\mathcal{L}_i = -\Big(y_i \log\left(a_i^{[2]}\right) + (1 - y_i) \log\left(1 - a_i^{[2]}\right)\Big). \tag{1.23}$$

In order to find suitable parameter values for $\mathbf{W}^{[1]}$, $\mathbf{b}^{[1]}$, $\mathbf{W}^{[2]}$, and $\mathbf{b}^{[2]}$, the gradient of the loss function, *i.e.*, $\mathcal{L}_i$, is minimized with respect to $\mathbf{W}^{[1]}$, $\mathbf{b}^{[1]}$, $\mathbf{W}^{[2]}$, and $\mathbf{b}^{[2]}$ to an extent where $\hat{y}_i \approx y_i$. For this purpose, a gradient descent optimization algorithm is used to update the parameter values as below

$$\mathbf{W}^{[2]} \leftarrow \mathbf{W}^{[2]} - \alpha \frac{\partial \mathcal{L}_i}{\partial \mathbf{W}^{[2]}}, \tag{1.24}$$

$$\mathbf{b}^{[2]} \leftarrow \mathbf{b}^{[2]} - \alpha \frac{\partial \mathcal{L}_i}{\partial \mathbf{b}^{[2]}}, \tag{1.25}$$

$$\mathbf{W}^{[1]} \leftarrow \mathbf{W}^{[1]} - \alpha \frac{\partial \mathcal{L}_i}{\partial \mathbf{W}^{[1]}}, \tag{1.26}$$

$$\mathbf{b}^{[1]} \leftarrow \mathbf{b}^{[1]} - \alpha \frac{\partial \mathcal{L}_i}{\partial \mathbf{b}^{[1]}}, \tag{1.27}$$

where α is the learning rate for the gradient descent optimization algorithm. A reasonable value of α helps in streamlining the learning of the NN during the training phase. Equations (1.24) through (1.27) are repeated till the parameters in $\mathbf{W}^{[2]}$, $\mathbf{b}^{[2]}$, $\mathbf{W}^{[1]}$, and $\mathbf{b}^{[1]}$ result in a $\hat{y}_i$ that is comparable to y_i. The sequence of steps to follow in a backward pass for the NN under discussion is $\mathbf{a}^{[2]} \rightarrow \mathbf{z}^{[2]} \rightarrow \mathbf{a}^{[1]} \rightarrow \mathbf{z}^{[1]}$. A summary of the steps to be followed for the internal parameter optimization of the

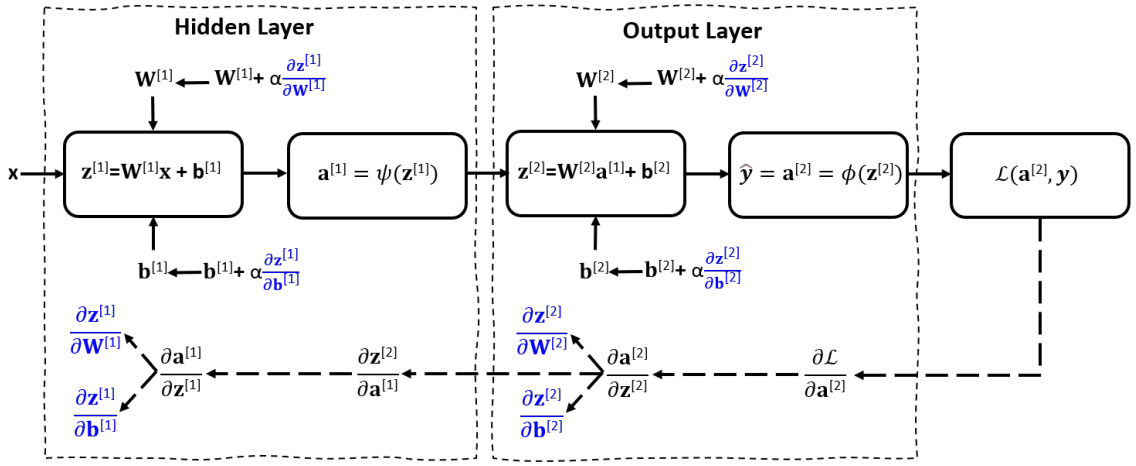

Figure 1.6: The sequence of steps followed in a feedforward neural network for optimization of the internal parameters.

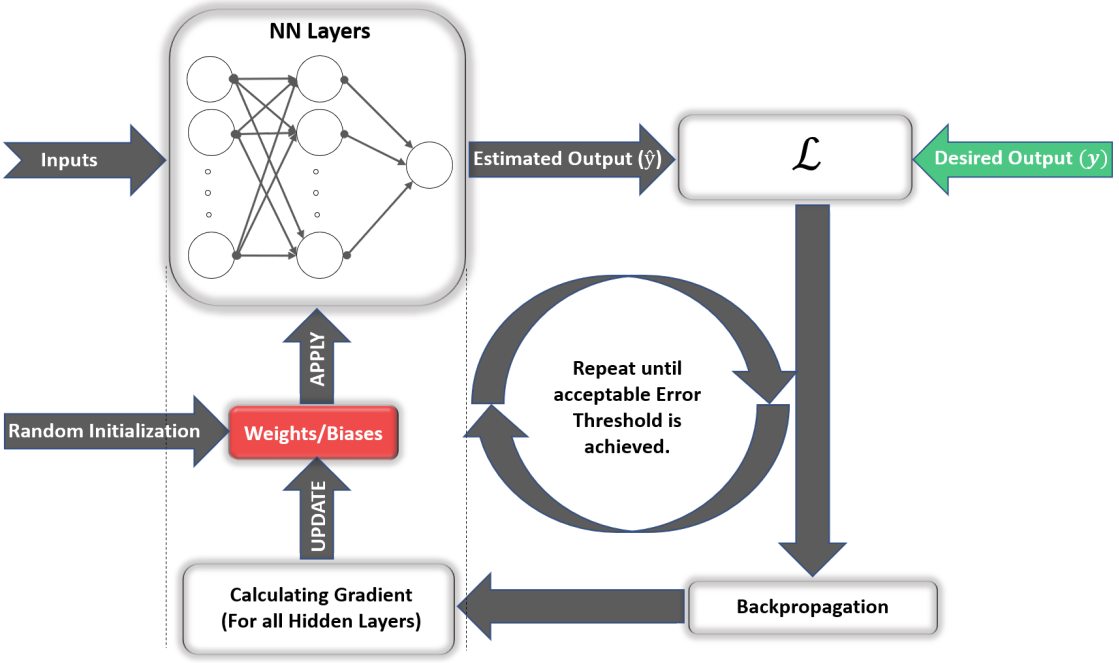

Figure 1.7: The internal process followed in a feedforward neural network for optimization of the internal parameters.

NN in forward and backward passes is shown in Figs. 1.6 and 1.7.

The terms $\frac{\partial\mathcal{L}_i}{\partial\mathbf{W}^{[2]}}$, $\frac{\partial\mathcal{L}_i}{\partial\mathbf{b}^{[2]}}$, $\frac{\partial\mathcal{L}_i}{\partial\mathbf{W}^{[1]}}$, and $\frac{\partial\mathcal{L}_i}{\partial\mathbf{b}^{[1]}}$ in eqs. (1.24)–(1.27) are given below

$$\frac{\partial\mathcal{L}_i}{\partial\mathbf{W}^{[2]}} = (a_i^{[2]} - y_i)\,\mathbf{a}^{[1]}. \tag{1.28}$$

$$\frac{\partial\mathcal{L}_i}{\partial b^{[2]}} = a_i^{[2]} - y_i. \tag{1.29}$$

$$\frac{\partial \mathcal{L}_i}{\partial \mathbf{W}^{[1]}} = \nabla z_i^{[2]}\, \mathbf{W}^{[2]} \left(1 - \left(\psi(\mathbf{z}_i^{[1]})\right)^2\right) \mathbf{x}. \tag{1.30}$$

$$\frac{\partial \mathcal{L}_i}{\partial \mathbf{b}^{[1]}} = \nabla z_i^{[2]}\, \mathbf{W}^{[2]} \left(1 - \left(\psi(\mathbf{z}_i^{[1]})\right)^2\right). \tag{1.31}$$

A step by step proof to eqs. (1.28)–(1.31) is given in Appendix A.

1.10 Analysis based on Real-world Experimental Data

This thesis not only validates claims made in various chapters through simulated data, but also takes into account real-world data collected in a 200 square meters field for numerical analysis. The real-world data comprises the RSS of the vehicles' transmitted signals in the field and their claimed locations. RSS measurements are preferred over ToA and AoA measurements as collecting the former is relatively straightforward compared to collecting the latter. In addition, recording ToA and AoA measurements require sophisticated equipment, which can be costly.

Three fixed RSUs independently record the RSS measurements of the vehicles' transmitted signals at a rate of one RSS measurement per RSU per second. The RSUs also receive the claimed locations of the vehicles at the same frequency. The RSS measurements and the claimed locations have associated timestamps with them. These timestamps help each RSU in combining the RSS measurements and the vehicles' claimed locations. The collected data is supplied to traditional LVSs and the newly designed intelligent ML-based LVSs for performance analysis. More details of numerical analysis involving the real-world data are available in Chapter 2 and Chapter 4.

1.11 Thesis Outline and Contributions

A brief overview of the rest of the chapters in this thesis is outlined next. In Chapter 2, the location of a vehicle will be acquired using NNs. An information-theoretic bound will be utilized to dive deep into the internal architecture of the designed

NN. The rest of the chapters will evolve around the verification of the acquired location information. Chapter 3 and Chapter 4 will integrate NNs into the area of location verification, showing (through simulated and experimental data) how NN-based LVSs will be able to overcome a few of the critical design assumptions of the information-theoretic LVSs. The latter chapters (*i.e.*, Chapter 5 and Chapter 6) will focus on practical steps to improve the performance of the designed NN-based LVS frameworks. These steps will revolve around optimizing the NNs' training time and classification efficiency. Chapter 7 will conclude this thesis with directions to future work. More specifically, the main contributions in each chapter are detailed below.

In Chapter 2, I will first acquire the location information of a user using NNs. I will then address an open problem related to the architecture of a Neural Network-based Location Estimation Framework (NNLEF). In particular, I will show how the Cramer-Rao Bound (CRB) on localization accuracy can facilitate in intelligently choosing the number of hidden layer neurons for the NNLEF. I will validate this new approach to formulate the internal architecture of the NNLEF through simulated and experimental RSS data. The main contents of this chapter has resulted in the following publication

- U. Ihsan, R. Malaney, and S. Yan, "Neural network architecture and location estimation in the Internet of Things," in *Proceedings of the IEEE International Communication Conference (ICC)*, Montreal, QC, Canada, Jun. 2021, pp. 1-6.

In Chapter 3, I will formulate a Neural Network-based Location Verification System (NN-LVS) to address a critical design limitation of the information-theoretic LVSs, *i.e.*, their idealistic assumptions on *a-priori* information on the proportions of genuine and malicious users in the field. The improved performance of this new form of LVS will be demonstrated based on simulated ToA measurements from multiple verifying BSs within the context of vehicular networks, quantifying how the designed NN-LVS can outperform a standalone information-theoretic LVS in a range of anticipated real-world conditions. The main contents of this chapter has resulted in the following publication

- U. Ihsan, R. Malaney, and S. Yan, "Artificial intelligence and location verification in vehicular networks," in *Proceedings of the IEEE Global Communication Conference (GlobeCom)*, HI, USA, Dec. 2019, pp. 1-6.

In Chapter 4, I, through experimental data, will quantify that the NN-LVS (designed in Chapter 3) is more realistic and practically deployable. For the first time, real-world RSS measurements from random vehicles will be used to show how the NN-LVS can outperform an information-theoretic LVS. The RSS measurements will be preferred over the other physical layer properties (i.e., ToA and AoA) of the transmitted signals in this chapter as the former can be collected more efficiently compared to the latter in real-world situations. The main contents of this chapter has resulted in the following publication

- U. Ihsan, R. Malaney, and S. Yan, "Machine learning and location verification in vehicular networks," in *Proceedings of the 8th IEEE/CIC International Conference on Communications in China (ICCC2019)*, Changchun, China, Aug. 2019, pp. 91-95.

In Chapter 5, I will consider a newly derived information-theoretic bound on the Total Error for an LVS to limit the training of the NN-LVS. This new bound will allow for a useful trade-off in learning-time *vs.* verification-performance for the NN-LVS. The main contents of this chapter has resulted in the following publication

- U. Ihsan, S. Yan, and R. Malaney, "Location verification for emerging wireless vehicular networks," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10261-10272, Aug. 2019.

In Chapter 6, I will design a joint Neural Network Framework (NNF) by combining multiple standalone NN-LVSs and show how the NNF can beat the best performance of the standalone NN-LVSs under all threat situations. The main contents of this chapter has resulted in the following publication

- U. Ihsan, and R. Malaney, "Combining different neural networks for location

verification," Submitted to: *IEEE International Communication Conference (ICC)*.

Chapter 7 will conclude this thesis and provide pointers to direction for future work.

Besides the regular Chapters, this thesis includes a number of appendices at the end. Together with the main chapters, these appendices will add to a reader's understanding of numerous technical concepts in the thesis. Appendix C and D are the contributions of my collaborators.

Chapter 2

Neural Network Architectures for Location Estimation in the Internet of Things

Tremendous efforts have been carried in the past and are still underway to formulate the architecture of an NN framework. To the best of our knowledge, little has been achieved in this regard so far. This chapter attempts to justify an NNLEF's architecture through information theory. How the number of neurons in the hidden layer of the NNLEF can be chosen is primarily investigated. Another aspect of the NN architecture that is covered here is how to decide the number of hidden layers and the transfer function in each layer of the NNLEF. Through simulated and real-world experimental RSS data, it is shown how the NNLEF performs efficiently once the number of neurons in the hidden layer is considered as per the recommendations. Although the NN architecture investigated in this chapter is in the context of a localization system, the concepts discussed here are believed to be equally useful for NN-based LVSs.

2.1 Introduction

Several traditional localization algorithms have been developed to estimate the location of a vehicle in the past, e.g., [45, 47, 89–91]. However, the practical limitations with these algorithms may range from limited functionalities to a complete failure as the surrounding environment changes. Therefore, we need localization algorithms that are practically deployable, smart enough to adapt to environmental changes, and are realistic.

To address the challenges that traditional localization algorithms face, researchers in recent times have incorporated numerous NN techniques for positioning users/devices [92, 93]. While NNs have been able to address the shallow learning capabilities of the classic ML algorithms, a key question about NNs, that is yet to be answered is how to design their internal architecture, *i.e.*, the number of chosen hidden layers, the choice of transfer function(s) in the hidden layer(s), and the number of neurons in each hidden layer. While the research community follows hyperparameters and other search mechanisms to finalize the architecture for NN frameworks at large [87, 94], numerous guidelines have also been provided in the recent literature to formulate an optimal NN architecture [88, 95]. However, to-date there has been no concrete solution on how to pre-determine an NN architecture for a given problem.

This chapter develops a feedforward NN framework for location estimation and formulates insight into its architecture. RSS measurements of the vehicles' transmitted signals recorded at multiple static RSUs are used as inputs. Through an analysis based on the CRB on the location accuracy of a vehicle, an architecture for an NNLEF is justified. Detailed numerical results confirm our analysis.

Beyond the contribution stated above, additional contributions in this chapter are summarized thus:

1. A value for the number of neurons needed in the hidden layer is derived.

2. Through simulated data, an efficient performance for the NNLEF (with the adopted architecture) is shown when compared to other NNLEFs (that follow

random architectures).

3. It is also shown how the NNLEF (with the adopted architecture) outperforms when compared to a traditional RSS-based algorithm.

4. Finally, the recommended architecture for the NNLEF is experimentally validated.

The remainder of this chapter is organized as follows. Section 2.2 details the system model and the derivation of the CRB on location accuracy. Section 2.3 presents the NNLEF. Section 2.4 provides numerical results based on simulated and experimental data, and Section 2.5 concludes this paper.

2.2 System Model and RSS Location Estimation

The following system model is taken into account in this chapter:

1. The true location of a random vehicle (which is unknown to the framework) is denoted by $\mathbf{x}_t = [\mathrm{x}_0, \mathrm{y}_0]$.

2. The framework has N number of RSUs with publicly known locations. The true location of the i-th RSU is $\mathbf{x}_i = [\mathrm{x}_i, \mathrm{y}_i]$ where $i = 1,\ 2,\ 3, \ldots, N$.

3. All the RSUs are in the transmission range of the randomly located vehicle and independently measure RSS (all RSS in dBm) of the transmitted signal (from the random vehicle) every second. A log-normal shadowing model is adopted for the RSS observations. The measured RSS at the i-th RSU, i.e., r_i, is given as

$$r_i[dBm] = P_T[dBm] - PL_{d_i}[dB], \tag{2.1}$$

where P_T is the transmit power of the vehicle and PL_{d_i} (the path loss at a distance d_i) is given by

$$PL_{d_i}[dB] = PL_{d_o} + 10\gamma \log_{10}\left(\frac{d_i}{d_0}\right) + X_{\sigma_{db}},\ \ (i = 1,\ 2,\ 3, \ldots, N), \tag{2.2}$$

where PL_{d_0} is the reference path loss at a reference distance d_0, γ is the path loss exponent, d_i is the vehicle-RSU$_i$ distance ($d_i > d_0$) given by $d_i = \sqrt{(\mathrm{x}_i - \mathrm{x}_0)^2 + (\mathrm{y}_i - \mathrm{y}_0)^2}$, and $X_{\sigma_{db}}$ is a zero-mean normal random variable with variance σ_{db}^2 representing the shadowing noise. The RSS measurements made by the N RSUs are independent of each other. They collectively form an RSS vector given by $\mathbf{r} = [r_1,\ r_2,\ r_3, \ldots, r_N]^\top$.

4. One of the N RSUs is chosen as the PC. The PC accumulates its RSS measurements with the regularly collected RSS measurements from all surrounding RSUs. The PC further processes these measurements to estimate the location of the random vehicle. The estimated location of the vehicle is denoted by $\hat{\mathbf{x}}_e = [\hat{\mathrm{x}}_e, \hat{\mathrm{y}}_e]$.

2.2.1 Cramer-Rao Bound Derivation

This section details the derivation of the CRB[1] on the location accuracy of a vehicle. For a random transmitting vehicle, whose location $\mathbf{x}_t$ is unknown and whose RSS is measured at N RSUs, the distribution of the RSS takes the form (with few constant elements ignored)

$$-\ln f_{r_i|\mathbf{x}_t\mathbf{x}_i} = \frac{[r_i + \gamma\,(\frac{10}{\ln 10})\,\ln(\frac{d_i}{d_0})]^2}{2\sigma_{db}^2}. \tag{2.3}$$

The covariance matrix $\mathcal{C}$ (related to the position) can be written as the inverse of the Fisher information matrix, $\mathcal{F}$, i.e., $\mathcal{C} = \frac{1}{\mathcal{F}}$. The elements of $\mathcal{F}$ are given by

$$\mathcal{F} = \begin{bmatrix} \mathcal{I}_{\mathrm{xx}} & \mathcal{I}_{\mathrm{xy}} \\ \mathcal{I}_{\mathrm{yx}} & \mathcal{I}_{\mathrm{yy}} \end{bmatrix}, \tag{2.4}$$

where

$$\mathcal{I}_{\mathrm{xx}} = -\mathbb{E}\left[\frac{\partial^2}{\partial \mathrm{x}_i\, \partial \mathrm{x}_i}(\ln f_{r_i|\mathbf{x}_t\mathbf{x}_i})\right], \tag{2.5}$$

$$\mathcal{I}_{\mathrm{xy}} = -\mathbb{E}\left[\frac{\partial^2}{\partial \mathrm{x}_i\, \partial \mathrm{y}_i}(\ln f_{r_i|\mathbf{x}_t\mathbf{x}_i})\right], \tag{2.6}$$

[1]More information on the CRB can be found in Appendix B.

$$\mathcal{I}_{\mathrm{yx}} = -\mathbb{E}\left[\frac{\partial^2}{\partial \mathrm{y}_i \, \partial \mathrm{x}_i}(\ln f_{r_i|\mathbf{x}_t\mathbf{x}_i})\right], \tag{2.7}$$

$$\mathcal{I}_{\mathrm{yy}} = -\mathbb{E}\left[\frac{\partial^2}{\partial \mathrm{y}_i \, \partial \mathrm{y}_i}(\ln f_{r_i|\mathbf{x}_t\mathbf{x}_i})\right], \tag{2.8}$$

where $\mathbb{E}$ denotes the expectation operation. The expressions in brackets are given as

$$\frac{\partial^2}{\partial \mathrm{x}_i \, \partial \mathrm{x}_i} = \frac{a\,\gamma}{\sigma^2_{db}} \sum_{i=1}^{N} \frac{1}{d_i^2}\left[\frac{a\,\gamma\,(\mathrm{x}_i - \mathrm{x}_0)^2}{d_i^2} + \Big(r_i + a\,\gamma\,\ln(d_i)\Big)\Big(1 - \frac{2\,(\mathrm{x}_i - \mathrm{x}_0)^2}{d_i^2}\Big)\right], \tag{2.9}$$

$$\frac{\partial^2}{\partial \mathrm{y}_i \, \partial \mathrm{y}_i} = \frac{a\,\gamma}{\sigma^2_{db}} \sum_{i=1}^{N} \frac{1}{d_i^2}\left[\frac{a\,\gamma\,(\mathrm{y}_i - \mathrm{y}_0)^2}{d_i^2} + \Big(r_i + a\,\gamma\,\ln(d_i)\Big)\Big(1 - \frac{2\,(\mathrm{y}_i - \mathrm{y}_0)^2}{d_i^2}\Big)\right], \tag{2.10}$$

$$\frac{\partial^2}{\partial \mathrm{x}_i \, \partial \mathrm{y}_i} = \frac{\partial^2}{\partial \mathrm{y}_i \, \partial \mathrm{x}_i} = \frac{a\,\gamma}{\sigma^2_{db}} \sum_{i=1}^{N} \frac{(\mathrm{x}_i - \mathrm{x}_0)\,(\mathrm{y}_i - \mathrm{y}_0)}{d_i^4}\left[a\,\gamma - 2\Big(r_i + a\,\gamma\,\ln(d_i)\Big)\right], \tag{2.11}$$

where $a = \frac{10}{\ln(10)}$.

Considering $d_0 = 1m$ and after some calculus, the final expressions for the elements of $\mathcal{F}$ are extracted as

$$\mathcal{I}_{\mathrm{xx}} = \frac{a^2\gamma^2}{\sigma^2_{db}} \sum_{i=1}^{N} \frac{(\mathrm{x}_i - \mathrm{x}_0)^2}{d_i^4}, \tag{2.12}$$

$$\mathcal{I}_{\mathrm{yy}} = \frac{a^2\gamma^2}{\sigma^2_{db}} \sum_{i=1}^{N} \frac{(\mathrm{y}_i - \mathrm{y}_0)^2}{d_i^4}, \tag{2.13}$$

$$\mathcal{I}_{\mathrm{xy}} = \mathcal{I}_{\mathrm{yx}} = \frac{a^2\,\gamma^2}{\sigma^2_{db}} \sum_{i=1}^{N} \frac{(\mathrm{x}_i - \mathrm{x}_0)\,(\mathrm{y}_i - \mathrm{y}_0)}{d_i^4}. \tag{2.14}$$

The final expression for CRB can be written as, $\rho^2 = \rho^2_{\mathrm{xx}} + \rho^2_{\mathrm{yy}}$. Here ρ^2_{xx} and ρ^2_{yy} are the diagonal elements of $\mathcal{C}$, and $\sqrt{\rho^2}$ equals the standard deviation of the CRB.

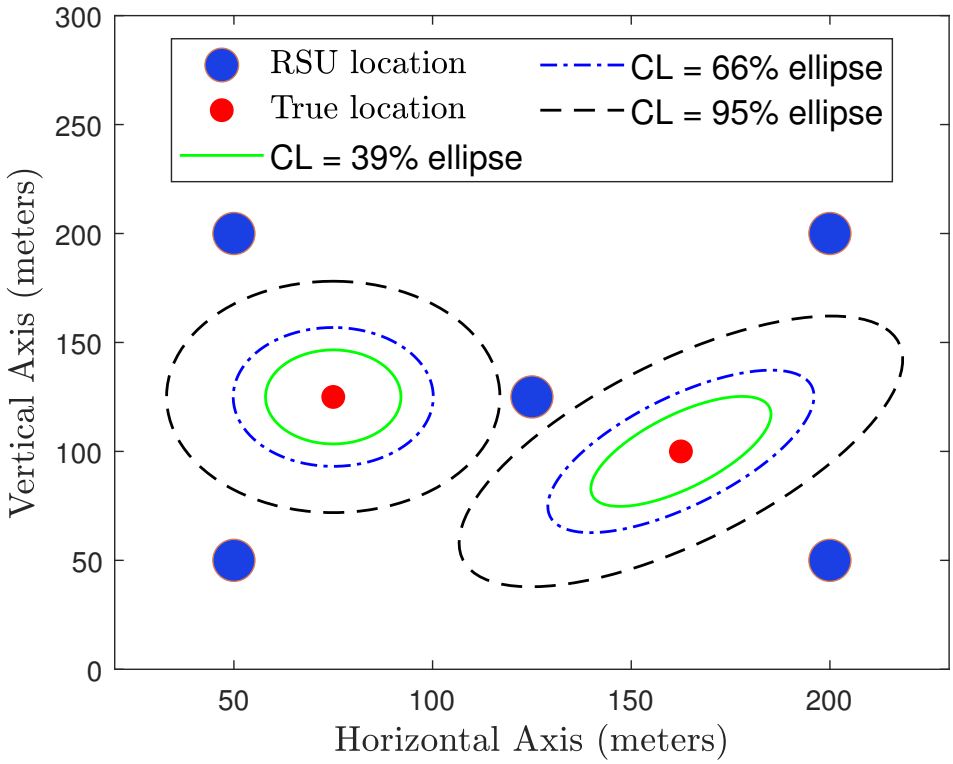

Figure 2.1: Ellipses with CLs of 39% (in solid green), 66% (in dashed blue), and 95% (in dashed black) are plotted for 2 sample test points. The test points in red form the origin of the respective ellipses.

To draw the ellipses representing the CRB on location accuracy in terms of Confidence Levels (CLs), the rotation matrix $\mathcal{R}$ is required and is given by

$$\mathcal{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \tag{2.15}$$

where $\theta = tan^{-1}(\lambda_1/\lambda_2)$, while λ_1 and λ_2 are the eigenvectors of $\mathcal{C}$. The range of θ is between 0 and 2π. The probability of a vehicle's location (returned by a positioning system) lying within a CL ellipse is given below [96]

$$P_{in} = 1 - e^{-\frac{K}{2}}, \tag{2.16}$$

where K is a constant that sets the scaling of the confidence ellipse.

In Fig. 2.1, ellipses with different CLs for two sample points are plotted using the derived equations. The locations of the RSUs are (50m, 50m), (50m, 200m),

(125m, 125m), (200m, 50m), and (200m, 200m). The value of σ_{db} is fixed at 5dB, and the path loss exponent equals 3.

2.3 Neural Network-based Location Estimation

This section highlights the adopted NNLEF's performance in estimating a vehicle's location. A feedforward NN forms the basis of this framework. This framework utilizes the measured RSS (influenced by the channel noise) at multiple RSUs. A feedforward network is a special type of NN known to manipulate and learn from the physical layer properties of the vehicles' transmitted signals [28, 97, 98]. NN frameworks with a single or multiple hidden layers can converge to a continuous target function. However, a single hidden layer NN framework has a more flexible learning rate and converges faster to the target function when compared to a multiple hidden layers NN framework [99]. The architecture for the NN framework in this chapter is thus limited to a single hidden layer.

Next, intuition is developed into the performance of the NNLEF with changing transfer functions and with a varying number of neurons in the hidden layer. It is evident from the expressions of a logistic sigmoid transfer function, i.e., $a(x) = (1 + e^{-x})^{-1}$, and a tangent sigmoid transfer function, i.e., $a(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$, that the gradient for both these transfer functions at absolute high values is approximately zero. With complex data at the input, this phenomenon can minimize learning for the NNLEF. On the other hand, a steady gradient for the ReLU transfer function, $a(x) = max[0, x]$, keeps the framework's learning consistent in the region where $x > 0$ and there is a possibility of faster learning for the NNLEF [100, 101]. Upon investigation, it has been found that different transfer functions in the hidden layer of the NNLEF produce comparable results. In order to accommodate for future complex channel environments, a ReLU is taken into account as the choice of transfer function for the hidden layer of the NNLEF.

The input to the NNLEF comprises $\mathbf{r}$. The number of outputs is set to two (the location coordinates). A schematic of the NNLEF in this chapter is shown

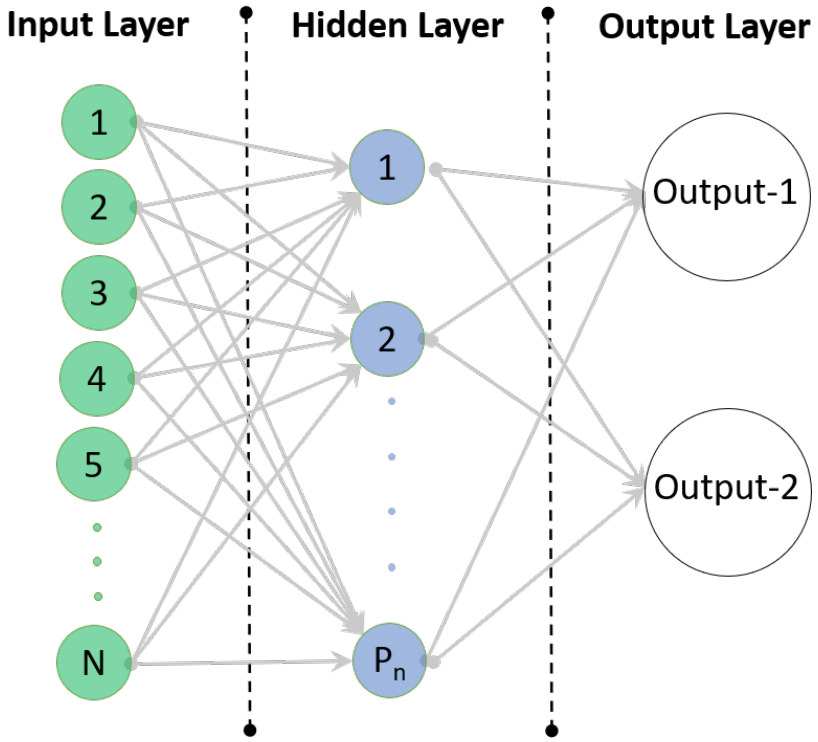

Figure 2.2: A schematic of the feedforward NN adopted for the NNLEF in this chapter. The number of inputs is set to N. The hidden layer has P_n neurons. The number of outputs in the output layer is set to 2, i.e., the dimension of a pair (of coordinates).

in Fig. 2.2. The number of neurons in the hidden layer, P_n, is our focal point in this study. The text in the following paragraphs will provide an insight into a recommendation for P_n. This recommendation for P_n will allow for a promising performance of the NNLEF. Each neuron in the hidden layer of the NNLEF partially contributes towards the performance of the NNLEF. A very small P_n is likely not sufficient to extract the hidden features/patterns in the input data. A reasonable P_n is, therefore, required for the NNLEF to perform efficiently. Increasing P_n is expected to produce good results (Figs. 2.3 and 2.5 highlight this phenomenon) but is not, in general, advised as this only adds to the framework's overhead. That is, increasing P_n can improve the NNLEF's performance marginally, but at the cost of an unnecessary increase in the NNLEF's overhead (e.g., the number of parameters, computational time, and memory resources). Additionally, with a very high P_n, the NNLEF can lead to an over-fitting problem, especially in conjunction when too much training data is supplied under one specific channel condition.

Two critical questions are faced in the design of any NN algorithm. One is: How much training of the algorithm should occur? A second is: What should the architecture be (how many neurons)? Consider a channel that is perfectly described by some model, and this is used for training purposes. If we were to train a NN under this model with unconstrained training samples (RSS values and all location information) and unconstrained P_n (the number of neurons to be placed in the hidden layer), we would be identifying perfectly, in effect, the function that describes the model's distance *vs.* RSS relation. Given this perfect channel identification, any use of that network to determine an unknown location from noisy RSS values should achieve a location error at the CRB.[2]

Carrying out the calculation in view of the analysis provided in Appendix D, taking a typical distance scale of order 100m, the following is found: Adopting $n = 3$ and $\sigma_{db} = 3$ result in $P_n = 12$; adopting $n = 3$ and $\sigma_{db} = 5$ result in $P_n = 9$; and adopting $n = 3$ and $\sigma_{db} = 8$ result in $P_n = 7$; This indicates P_n in the range of 7-12 would be useful for the type of channels under investigation here.

2.4 Numerical Results

2.4.1 Analysis Using Simulated Data

This section first presents the numerical results based on simulated RSS data. The focus area is of size 200m×200m. 5 RSUs are installed at (0m, 0m), (0m, 200m), (100m, 100m), (200m, 0m), and (200m, 200m). The horizontal and vertical axes are partitioned into equidistant divisions. The RSUs measure RSS from the cross section of the divisions on both the axes at a frequency of 1Hz. The RSS measurements are under the influence of random shadowing noise. To mimic reality and accommodate for the unique location of each RSU, this noise element is extracted from a random Gaussian distribution with a fixed σ_{db}. This means that at any given instant, the

[2]However, in real-world scenarios, we could expect the location accuracy of such a network be substantially less than the CRB. The reason for this is that, in general, the real-world channel will never be exactly the training model.

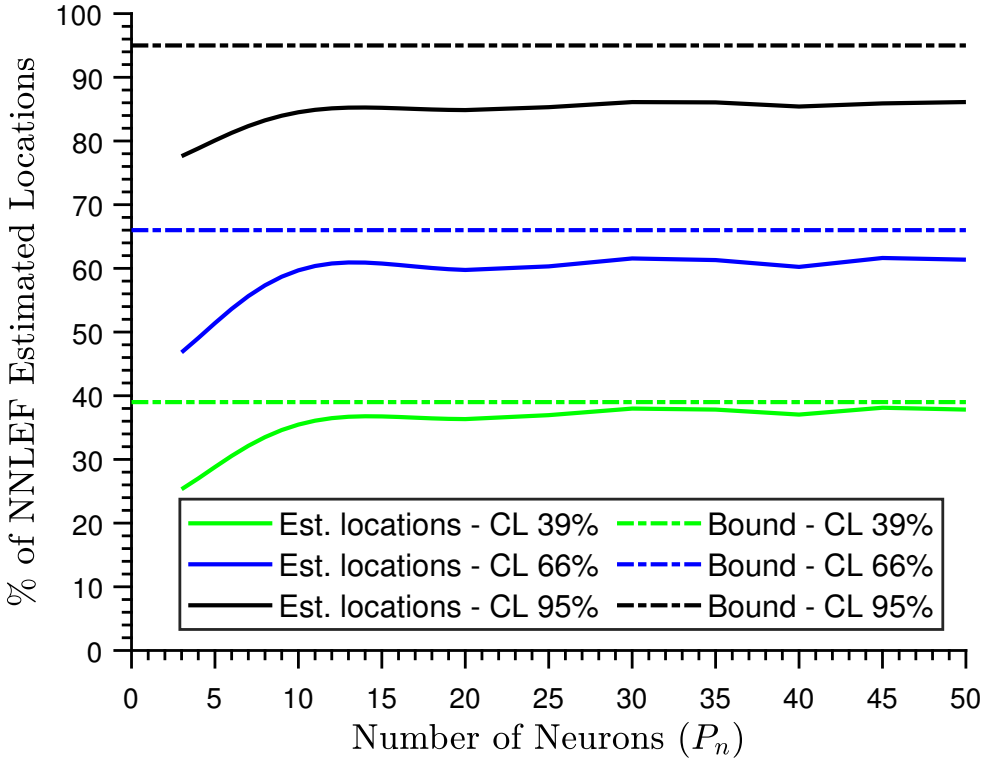

Figure 2.3: The percentage of estimated test locations by NNLEFs with a changing P_n. The number of RSUs is 5, and the value of σ_{db} is set to 5dB. The horizontal axis shows the number of neurons in the hidden layer, P_n, for different NNLEFs. The solid green, blue, and black lines represent the percentage of estimated locations for vehicles in ellipses with CLs 39%, 66%, and 95%, respectively. The dashed lines indicate the CRBs on location accuracy for the corresponding confidence ellipses. It is evident that the performance for the NNLEF becomes nearly steady as P_n equals or exceeds 8.

RSS measurements from a cross section of the divisions on all the RSUs will have unique and independent shadowing noise elements included in them. A value of 5dB is taken into account for σ_{db}. The path loss exponent is set to 3. After the RSS measurement campaign, the RSS database is randomized and further divided into two sets; a test set (with nearly 10% of the database samples) and a training set (with the remaining database samples). The training set has the horizontal and vertical coordinates for the cross section of the divisions, while the test set has no such information included.

Fig. 2.3 examines the performance of the NNLEFs with changing P_n. For uniformity, a ReLU transfer function is used in the hidden layer of all the NNLEFs. Moreover, all the other NN training parameters are kept the same. All the NNLEFs

are trained with the same training data. Post training, the NNLEFs are subjected to estimate locations for the vehicles in the test set. To analyze the performance, confidence ellipses with different CLs and the estimated locations by each NNLEF are plotted. To determine whether an NNLEF's estimated location is within or outside a particular confidence ellipse, the following equation is utilized

$$\frac{(\cos(\theta)(\hat{\mathrm{x}}_e - \mathrm{x}_c) + \sin(\theta)(\hat{\mathrm{y}}_e - \mathrm{y}_c))^2}{l_{maj}} + \frac{(\sin(\theta)(\hat{\mathrm{x}}_e - \mathrm{x}_c) - \cos(\theta)(\hat{\mathrm{y}}_e - \mathrm{y}_c))^2}{l_{min}} \leqslant 1, \tag{2.17}$$

where $(\mathrm{x}_c, \mathrm{y}_c)$ is the center of the ellipse, l_{maj} is the length of the semi-major axis, and l_{min} is the length of the semi-minor axis for a particular confidence ellipse. In Fig. 2.3, the percentage of the NNLEFs' estimated test locations in each confidence ellipse (on the y-axis) is plotted against P_n (on the x-axis). The changing P_n on the x-axis corresponds to different NNLEFs. A polynomial fitting of order 7 is applied for curve smoothing. The green, blue, and black curves represent the percentage of estimated locations by the NNLEFs in ellipses with CLs 39%, 66%, and 95%, respectively. The dashed coloured lines represent the CRBs on location accuracy for the corresponding confidence ellipses (derived in section 2.2.1). From the figure, we observe that the performance for the NNLEF becomes approximately consistent when $P_n \geq 8$. We do see negligible performance improvement for a few random NNLEFs with P_n in the higher range. As such, these NNELFs are not recommended as they will increase the number of training parameters and computational costs by many folds, which do not justify the minimal performance improvement. For example, the number of training parameters for the NNLEF with 8, 30, 40, and 50 neurons in the hidden layer is 66, 242, 322, and 402, respectively.

Next, the performance for the NNLEF (with P_n in the recommended range, i.e., 8) is compared with a state-of-the-art RSS based algorithm. The RSS based algorithm minimizes the root mean square error between the measured RSS values in the test and training set to estimate a vehicle's location, i.e.,

$$\hat{\mathbf{x}}_e = min\left(\frac{1}{N}\sum_{m=1}^{M}(\mathbf{r}_{test} - \mathbf{r}_{train_m})\right), \tag{2.18}$$

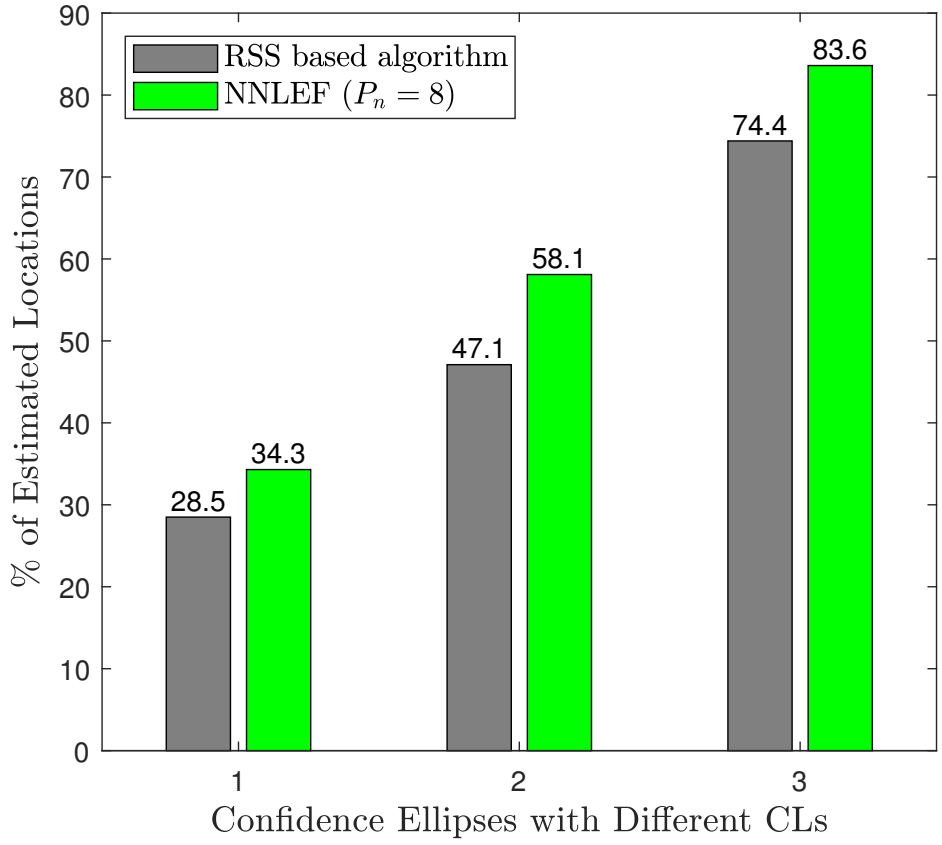

Figure 2.4: Performance comparison for the NNLEF (with $P_n = 8$) and a state-of-the-art RSS based algorithm. All the simulation parameters are identical to those used in Fig. 2.3. One can see that the NNLEF performs more efficiently when compared to the RSS based algorithm.

where $\mathbf{r}_{test}$, and $\mathbf{r}_{train}$ are the testing and training set RSS vectors, respectively, and M represents the total number of samples in the training set. Fig. 2.4 shows the percentage of the estimated test locations for the NNLEF and the RSS based algorithm in each confidence ellipse.

In order to further validate the performance for the NNLEF with the derived architecture, i.e., $P_n = 8$, in comparison to other NNLEFs with random architectures, i.e., with $P_n =$ equal to 3, 20, 30, 40, and 50, a different performance metric as in [93] is used, i.e., Mean Square Error (MSE). This new metric is mathematically defined as $\text{MSE} = \sqrt{(\hat{\text{x}}_e - \text{x}_0)^2 + (\hat{\text{y}}_e - \text{y}_0)^2}$. Using the same parameter settings as used in Fig. 2.3, the MSE for all the NNLEFs is plotted in Fig. 2.5. The x-axis indicates the MSE bin spacing in tens of meters, while the y-axis shows the number of samples in each MSE bin. The solid arrow pointing at the x-axis is the 1σ CRB in meters.

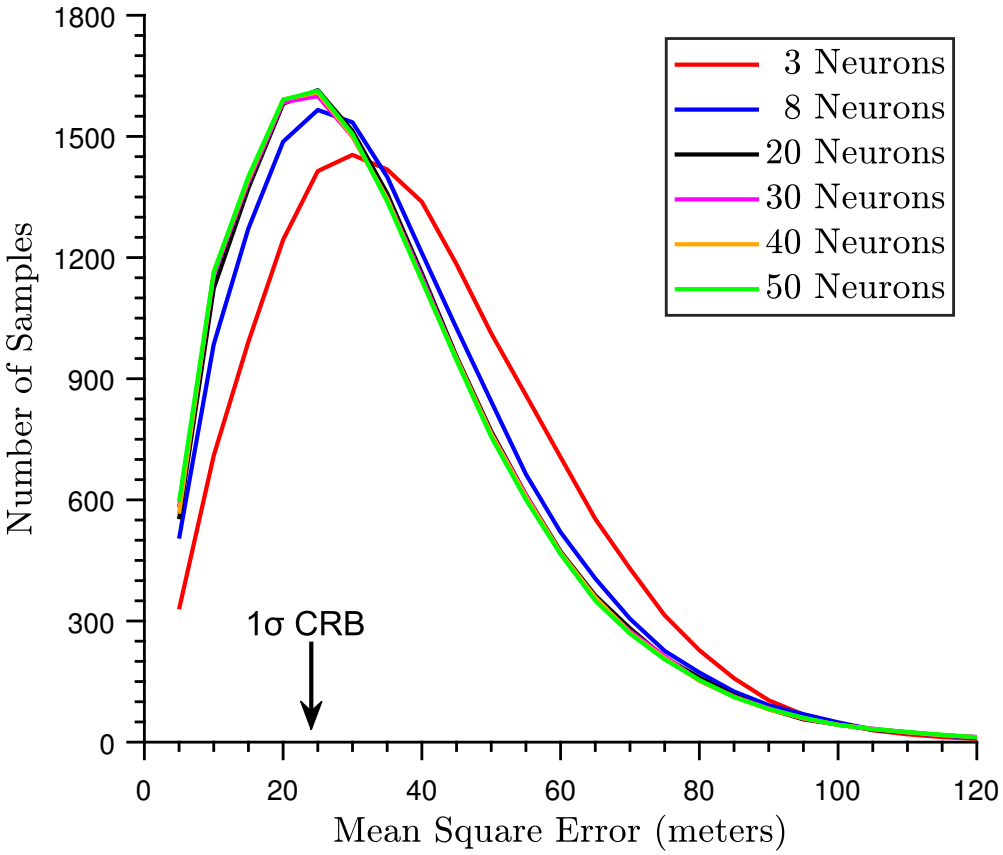

Figure 2.5: Performance evaluation for the NNLEF with $P_n = 8$ (in the recommended range) and other NNLEFs with random P_n. Similar parameter settings as used in Fig. 2.3 are taken into account, but a different performance evaluation metric, i.e., MSE (as in [93]) is considered. We see that NNLEF with $P_n = 3$ is under performing. Moreover, we observe an equal performance for the NNLEF with $P_n = 8$ and the other NNLEFs with higher P_n values.

We see an equivalent performance for all the frameworks. This highlights the fact that a high P_n does not always relate to the NNLEF's performance improvement.

2.4.2 Analysis Using Real-world Data

This section presents numerical results by taking into account real-world RSS measurements. These measurements have a multipath factor (from the ground) and noise elements in them. The RSS measurements from random vehicles were collected in a 150 X 150 square meters area by three RSUs (installed at (0m, 0m), (-25.5m, 47.6m), and (-8.6m, -46.9m)). Three devices were used to mimic three RSUs. Each device independently measured RSS from the random vehicles at a frequency of one

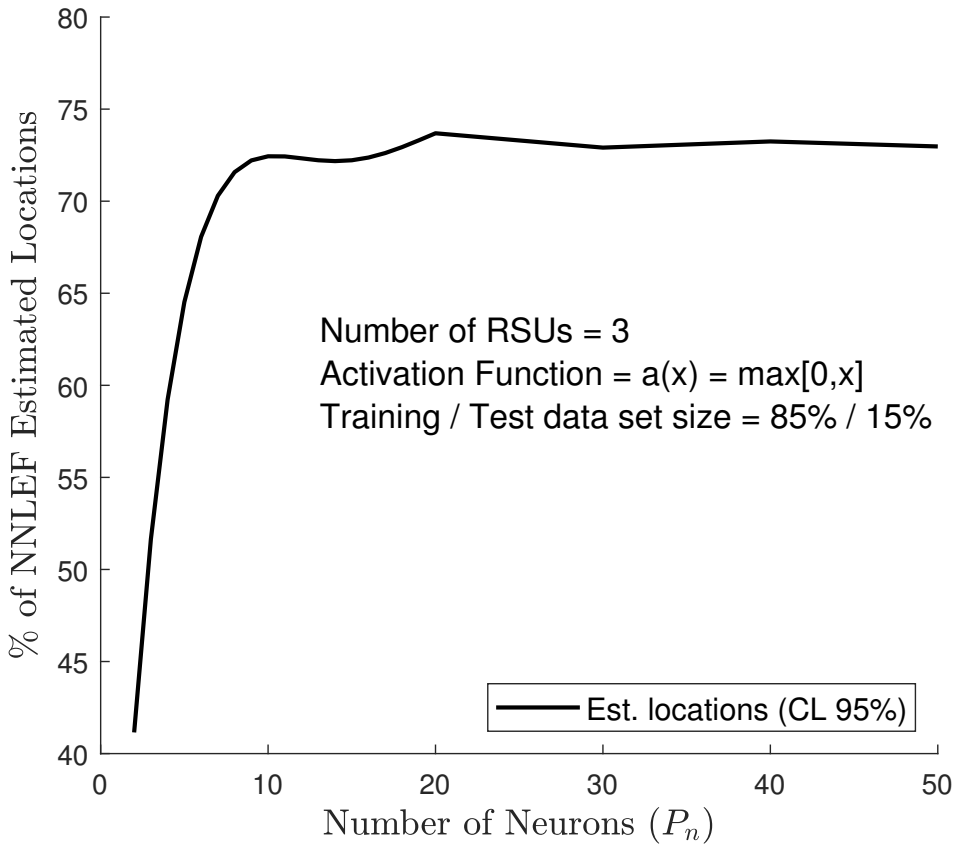

Figure 2.6: The percentage of estimated real-world test data locations by NNLEFs' (with changing P_n) in the ellipse with CL = 95%.

RSS measurement per second. Slowly moving Wi-Fi modems (802.11g) with a single antenna (at the same height as RSUs antennas), and a transmission frequency of 2.437 MHz were used to represent slow-moving vehicles. These vehicles, equipped with GPS units, reported their GPS locations to the RSUs every second. The RSS measurements at the individual RSUs and the GPS locations of the vehicles were combined utilizing the timestamps (available with both the RSS measurements and the vehicles' GPS locations).

At the end of the measurement campaign, the RSS measurement data was thoroughly randomized and divided into a training set (with 85% of the measurement data) and a test set (of the remaining 15% measurement data). The training set had the location information of the vehicles, while the test set had no such information included. All the NNLEFs (with different P_n) were trained using the training set. The trained NNLEFs' were then used to estimate the locations of the vehicles in

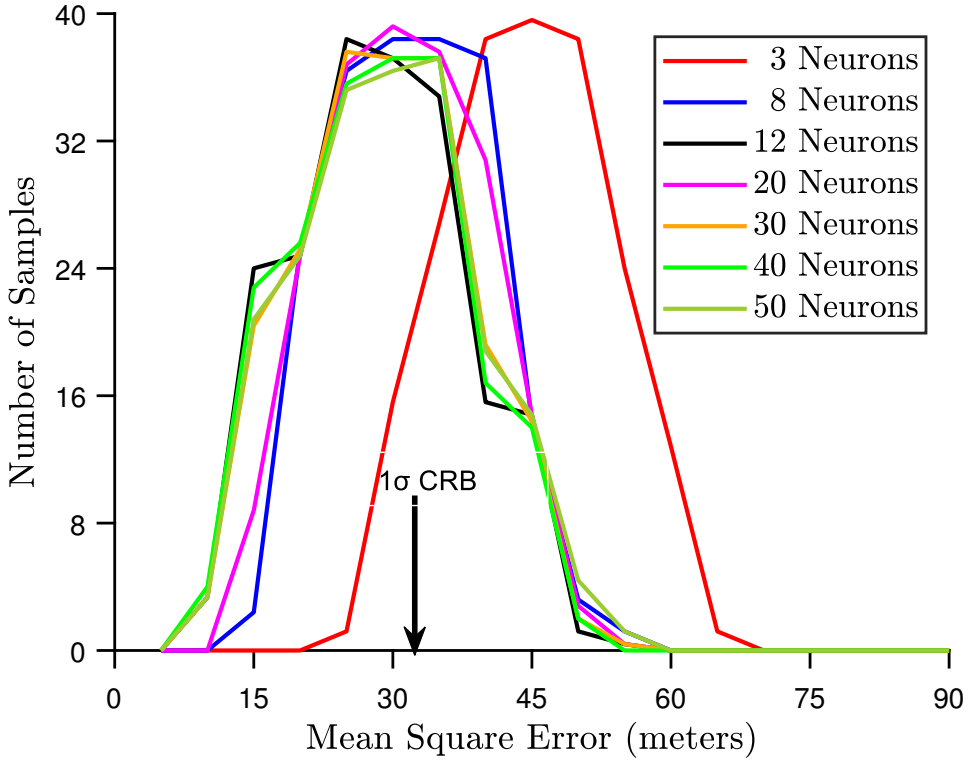

Figure 2.7: MSE performance evaluation for NNLEFs with P_n in the recommended range (i.e., P_n = 7 to 12) and other NNLEFs' with random P_n. We observe a poor performance for NNLEF with $P_n = 3$. On the other hand, we see an equivalent performance for the NNLEFs with the recommended P_n compared to the other NNLEFs with higher P_n.

the test set. Fig. 2.6 is a plot of the percentage of the estimated test set locations for NNLEFs with changing P_n in the ellipse with 95% CL (with polynomial fitting applied). From the figure, we observe that performance for the NNLEFs becomes asymptotic once P_n exceeds 8. This validates the claim made earlier that a higher P_n may not add much to the performance of the NNLEF. Instead, it would increase the computational overheads.

Fig. 2.7 is a comparison of the performance of an NNLEF with $P_n = 8$ and 12 (in line with the recommended range for P_n) and other NNLEFs with random P_n using the MSE metric. We notice that NNLEF with too low a P_n, i.e., 3, performs poorly. We also see nearly equivalent performance for the NNLEF (with $P_n = 8$ and 12 neurons) when compared to the other NNLEFs (with high P_n). Here, we observe that NNLEF with $P_n = 12$ performs slightly better than NNLEF with $P_n = 8$.

2.5 Conclusion

In this chapter, it has been shown how information-theoretic constructs can be used to decide the number of hidden-layer neurons within NN architectures for wireless location estimation. The analysis is confirmed through simulated as well as real-world data. Furthermore, the analysis in this chapter provides insight into pragmatic architecture design for a wide range of NN frameworks beyond location estimation.

In the next chapter, we will take advantage of the NNs to address a more critical problem, *i.e.*, verification of the acquired location information.

Chapter 3

Neural Networks and Location Verification in VANETs

In the previous chapter, the internal architecture of a feedforward NN for location estimation was investigated. In particular, the number of hidden layer neurons deemed sufficient to explore the channel type under consideration was formulated. We also looked into the choice of transfer functions for various layers of the feedforward NN. As claimed, a similar architecture will suffice for a framework aimed at verifying vehicles reported locations. This chapter considers the recommendations (from the previous chapter) in designing a novel NN for location verification.

Location information claimed by devices will play an ever-increasing role in future wireless networks such as wireless vehicular networks, 5G, and the IoT. Against this background, the verification of such claimed location information will be an issue of growing importance. However, the working of the already designed and available location verification frameworks is constrained by the assumptions made at the time of their design. One such important assumption for information-theoretic location verification frameworks is the prior knowledge on the proportion of malicious and legitimate users in the field. This chapter addresses this limitation by designing an NN-LVS. Working of this new form of LVS is free from any advance knowledge on the ratio of malicious or legitimate users in the field. Compared to the information-

theoretic LVS, this chapter also reports a more efficient performance for the NN-LVS when the ToA of the user's transmitted signals is under the influence of NLoS bias. The analysis in this chapter is based on simulated ToA measurements of the users' transmitted signals.

3.1 Introduction

We are on the verge of new wireless networks that aim to bring communication revolutions in homes, hospitals, education, transportation, and other aspects of society. Emerging ITS is particularly exciting due to its potential to save many lives. The success of these new technologies in general, and a VANET (a sub-application of ITS) in particular, find their roots in the accurate location information of the clients (e.g., devices, users, vehicles) involved. In many scenarios, it is anticipated that clients can directly obtain their location information [29, 50] through the GNSS. The clients usually provide this location information to other clients or to some central PC for network functionality purposes. But what if a client provides incorrect information about its true location intentionally in an attempt to obtain some advantage over other users [20, 21]? Such circumstances could also occur unintentionally due to difficulty in recording the GNSS location information or hardware issues.

The focus of the work in this chapter is location verification in the IoT, in general, and in VANETs in particular. The location verification framework proposed in this chapter is applicable to all IoT applications whose performance is related to a user's reported location. Within VANETs if a malicious vehicle provides inaccurate location information and this goes unnoticed, the possible aftermath could range from sub-optimal traffic routing all the way through to life-threatening collisions [1, 2]. Verification of a vehicle's reported location information is hence critical for successful operation in VANETs [1, 11–13, 102]. Due to this, LVS performance has been a research focus in VANETs for well over a decade. Recently, several information-theoretic LVSs have been devised [19, 22, 24, 81]. These LVSs operate under a set of well-defined rules and conditions. Additionally, they have limitations in addressing

various anomalies since they usually assume idealized channel conditions [22]. As such, information-theoretic LVSs usually possess performance limitations in real-world situations. One of the most important of these limitations is the *a-priori* lack of knowledge on the proportion (fraction) of vehicles in the field that will be malicious (alternatively, the fraction that will be legitimate). This issue is addressed in this chapter by designing an NN-LVS whose working is free from the knowledge on the proportions of malicious and legitimate vehicles. Compared to the information-theoretic LVSs, the new LVS shows promising performance in situations where the vehicles' transmitted signals have NLoS biases in them. Here, an LVS based on ToA measurements [23] under the influence of NLoS biases is considered. The novel contributions in this chapter are summarized below.

- It is shown that the NN-LVS proposed in this chapter outperforms an information-theoretic LVS [23] when the ToAs of the vehicles' signals have added NLoS bias in them.
- It is also shown that, unlike the information-theoretic LVS which assumes an *a-priori* knowledge about the proportion of malicious vehicles in the field, the NN-LVS works satisfactorily in the complete absence of this knowledge.

Recent advancements in digital signal processing and hardware design now provide us with very accurate physical-layer timing information for wireless networks [103, 104]. Furthermore, these developments provide us with the clock synchronization to enable the LVS that is studied here. As such, the new NN-LVS can offer a viable and pragmatic solution to the important task of location verification for many IoT applications under real-world conditions and uncertainties.

The remainder of this chapter is organized as follows. Section 3.2 presents the system model. Section 3.3 details the performance analysis using information theory and NN techniques. Section 3.4 provides numerical results, and Section 3.5 concludes this chapter.

3.2 System Model

The system model and assumptions considered in this chapter are outlined as below;

1. The system model consists of N trusted BSs as verifiers with publicly known locations that are assumed to be in the range of the prover (the vehicle whose claimed location is to be authenticated). The location of the i-th BS is $\mathbf{x}_i = [\mathrm{x}_i, \mathrm{y}_i]$ where, $i = 1, 2, 3, \ldots, N$.

2. The true location of a legitimate or malicious vehicle (the prover) is denoted by $\mathbf{x}_t = [\mathrm{x}_t, \mathrm{y}_t]$ and is assumed to possess zero localization error.

3. The announced (reported) location from a legitimate or malicious vehicle is referred to as the 'claimed location' and is denoted by $\mathbf{x}_c = [\mathrm{x}_c, \mathrm{y}_c]$. For a legitimate vehicle, the claimed location is precisely the same as its true location. On the other hand, a malicious vehicle spoofs its true location to the BSs (to potentially obtain an advantage over other vehicles or to disrupt the system performance). As a result, the true location of a malicious vehicle is unknown to the wider network.

4. One of the N BSs is chosen as the PC. Measurements from all BSs are collected at the PC before being processed into a binary decision related to a vehicle's claimed location.

5. Under the null hypothesis $\mathcal{H}_o$, the framework assumes a vehicle to be legitimate, *i.e.*,

$$\mathcal{H}_o : \mathbf{x}_c = \mathbf{x}_t. \tag{3.1}$$

6. Under the alternate hypothesis $\mathcal{H}_1$, the framework considers a vehicle to be malicious, *i.e.*,

$$\mathcal{H}_1 : \mathbf{x}_c \neq \mathbf{x}_t. \tag{3.2}$$

Under $\mathcal{H}_o$, the ToA value, y_i, measured by the i-th BS from a legitimate vehicle, is given by

$$y_i = u_i + z_i, \qquad i = 1, 2, 3, \ldots, N, \tag{3.3}$$

where z_i, the BS's receiver thermal noise, is a zero-mean normal random variable with variance σ_T^2. The variable u_i is the ToA of the vehicle's transmitted signal and is given by

$$u_i = \frac{d_i^c}{c}, \tag{3.4}$$

where d_i^c is the Euclidean distance of i-th BS to the legitimate vehicle's claimed location and is given by

$$d_i^c = \sqrt{(\mathrm{x}_c - \mathrm{x}_i)^2 + (\mathrm{y}_c - \mathrm{y}_i)^2}, \tag{3.5}$$

with c as the speed of light.

The measurements made by the N BSs are assumed to be independent of each other. Under $\mathcal{H}_o$, they collectively form a vector $\mathbf{y} = [y_1,\ y_2,\ y_3, \ldots, y_N]^\top$. The vector $\mathbf{y}$ follows a multi-variate normal distribution given as

$$\mathbf{y}|\mathcal{H}_o \sim \mathcal{N}(\mathbf{u}, \mathbf{R}), \tag{3.6}$$

where $\mathbf{u} = [u_1,\ u_2,\ u_3, \ldots, u_N]^\top$ is the mean vector under the null hypothesis, and $\mathbf{R} = \sigma_T^2 \mathbf{I}_N$ is the covariance matrix. Here, $\mathbf{I}$ represents an identity matrix.

Under $\mathcal{H}_1$, a malicious vehicle claims to be at a location removed from its true location. In a real-world scenario, we can think of this as if the malicious vehicle pretends to be on the road when he actually is placed off the road in a street. The ToA value measured by the i-th verifier from a malicious vehicle is given by

$$y_i = T_x + w_i + z_i, \qquad i = 1, 2, 3, \ldots, N, \tag{3.7}$$

where T_x is a time bias potentially added by the malicious vehicle that impacts the overall ToA value, and w_i is given by

$$w_i = \frac{d_i^t}{c}, \tag{3.8}$$

where d_i^t is the Euclidean distance from i-th BS to the malicious vehicle's true location, and is given by

$$d_i^t = \sqrt{(\mathrm{x}_t - \mathrm{x}_i)^2 + (\mathrm{y}_t - \mathrm{y}_i)^2}. \tag{3.9}$$

The measurements made by N BSs under $\mathcal{H}_1$ are also assumed to be independent of each other and collectively form a vector $\mathbf{y} = [y_1,\ y_2,\ y_3, \ldots, y_N]^\top$. The vector $\mathbf{y}$ follows a multi-variate normal distribution given as

$$\mathbf{y}|\mathcal{H}_1 \sim \mathcal{N}(\mathbf{w} + T_x\mathbf{1}, \mathbf{R}), \tag{3.10}$$

where $\mathbf{w} + T_x\mathbf{1} = [w_1 + T_x,\ w_2 + T_x,\ w_3 + T_x, \ldots, w_N + T_x]^\top$ and $\mathbf{1}$ is a vector equal to the length of the number of BSs, *i.e.*, N, with all its elements set to 1. For later convenience, $\mathbf{v} = \mathbf{w} + T_x\mathbf{1}$. The Eq. (3.10) is rewritten as

$$\mathbf{y}|\mathcal{H}_1 \sim \mathcal{N}(\mathbf{v}, \mathbf{R}). \tag{3.11}$$

3.3 Performance Analysis

Unlike a localization system, an LVS classifies the prover as either legitimate, or malicious at the output. The analysis for the LVS is carried out using two different methodologies in this chapter: through the information-theoretic procedures followed in [23], and through the newly designed NN method. In both cases, a Bayes average cost function is chosen as the performance metric for LVS in terms of the 'Total Error'. The information-theoretic method is based on the *a-priori* assumption that the proportion of malicious vehicles is known. Usually, this is set to 0.5 in the absence of any other information. On the other hand, the NN-LVS determines the

Total Error irrespective of any such *a-priori* assumption, and as will be shown, it can function with any proportion of malicious vehicles in the field. The Total Error is given by

$$\xi = P_0\alpha + P_1(1-\beta), \tag{3.12}$$

where P_0, and P_1 ($P_0 + P_1 = 1$) are the *a-priori* probabilities of occurrences of $\mathcal{H}_o$ (*i.e.*, legitimate vehicle), and $\mathcal{H}_1$ (*i.e.*, malicious vehicle), respectively, and are set equal, *i.e.*, 0.5. The variables α, and β represent the false positive rate (the rate of legitimate vehicles being detected incorrectly), and the detection rate (the rate of malicious vehicles being detected correctly), respectively. The Eq. (3.12) then takes the form

$$\xi = 0.5\alpha + 0.5\,(1-\beta)\,. \tag{3.13}$$

3.3.1 LVS Using Information Theory

It has been proven earlier that the LRT achieves the optimum detection results for a given false positive rate [105]. This leads to the conclusion that the LRT minimizes the Total Error and maximizes the mutual information between input and output of LVS [79]. The decision rule for the LRT is written as

$$\Lambda(\mathbf{y}) \triangleq \frac{p(\mathbf{y}|\mathcal{H}_1)}{p(\mathbf{y}|\mathcal{H}_o)} \underset{\mathcal{D}_0}{\overset{\mathcal{D}_1}{\gtrless}} \lambda, \tag{3.14}$$

where $\Lambda(\mathbf{y})$ is the likelihood ratio, $\lambda = \frac{P_0}{P_1}$ is the decision threshold, and $\mathcal{D}_1$ and $\mathcal{D}_0$ are the binary decision values (*i.e.*, whether the vehicle is malicious or legitimate). With the assumed proportions of legitimate and malicious vehicles, the variable $\lambda = 1$. Given the multi-variate normal form of the observations, the Eq. (3.14) can be reformulated as [25]

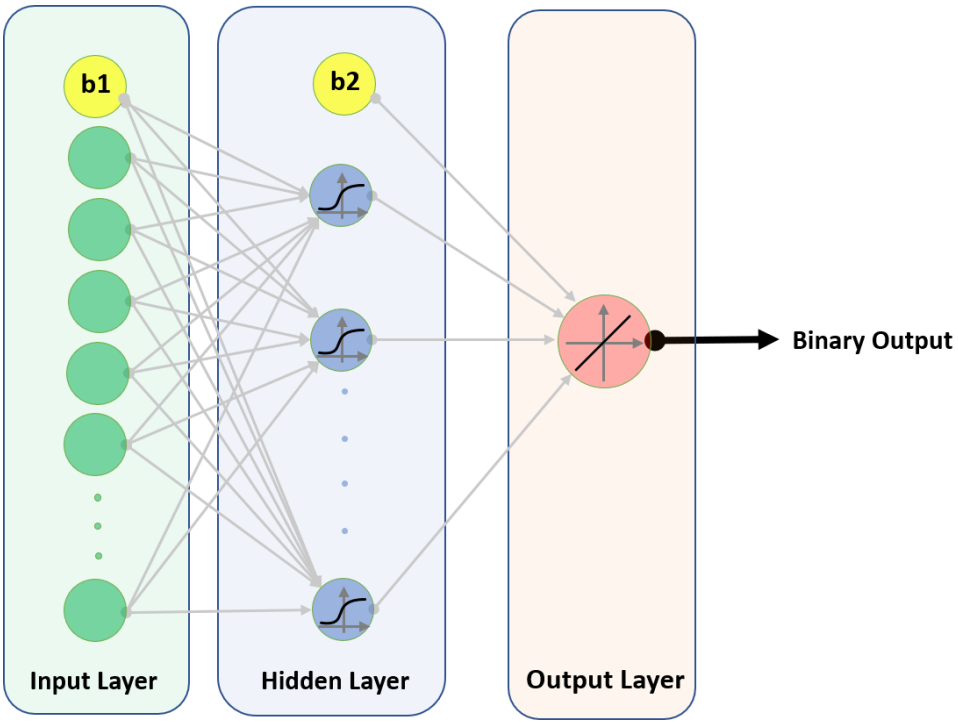

Figure 3.1: The architecture of the NN used for location verification in this chapter. This architecture arises from many trials of biased timings using different architectures with varying numbers of layers.

$$\Lambda(\mathbf{y}) = \frac{e^{-\frac{1}{2}(\mathbf{y}-\mathbf{v})^\top \mathbf{R}^{-1} (\mathbf{y}-\mathbf{v})}}{e^{-\frac{1}{2}(\mathbf{y}-\mathbf{u})^\top \mathbf{R}^{-1} (\mathbf{y}-\mathbf{u})}} \underset{\mathcal{D}_0}{\overset{\mathcal{D}_1}{\gtrless}} \lambda. \tag{3.15}$$

3.3.2 LVS Using Neural Networks

This section highlights the novel approach used to design a classification framework for the verification of a vehicle's claimed location through supervised NN techniques. The framework uses a multi-layer feedforward NN for the binary classification of a vehicle as either legitimate or malicious.

Building on the recommendations in the previous chapter, an NN-LVS with an input layer, a hidden layer (with ten neurons), and a binary output layer is finalized as shown in Fig. 3.1. For uniformity, the framework considers the same inputs as considered for the information-theoretic method. These inputs include the vehicle's claimed location and $\mathbf{y}$ (the observation vector influenced by the thermal noise z_i).

The NN-LVS achieved satisfactory performance through the use of the hyperbolic tangent sigmoid, and linear transfer functions in the hidden, and output layers, respectively.

3.4 Numerical Results

This section presents some numerical results based on the information-theoretic and NN-LVS analysis. In carrying out these simulations, BSs are located in a 1000 meters by 500 meters area at fixed publicly known locations. This area closely resembles a small district of a city and corresponds to the context of a VANET where the BSs are trusted verifiers located on the roadside or in the nearby parking lots. The claimant vehicle (the prover whose location is yet to be verified) resides in a 500 X 500 meters area in between the BSs. In order to simulate the attacking scenario and thus study the performance of both the LVSs, two claimants are assumed within the BSs' communication range, namely a legitimate vehicle which is reporting its true location to the BSs and a malicious vehicle that is performing the location-spoofing attack. Both the vehicles can overhear the communication between the BSs, and thus both acquire the locations of the BSs. If malicious, the vehicle can also overhear the communication between legitimate vehicles and the BSs so that it can forge its claimed location to that of the legitimate vehicle's true location.

The malicious vehicle sets its true location at a far-off point so that it's transmitted signal (with the appropriate timing offset) has equal ToA at all the BSs (in the limit of the true location of a malicious vehicle being much greater that any other scale all NLoS biases at all BSs are the same). Under this approximation, the mean ToA at the BSs is just the mean of the timings anticipated from a vehicle at the claimed location. The resultant alteration in ToA due to the receiver's thermal noise is extracted from a Gaussian random distribution with a fixed standard deviation. The value of standard deviation considered in the simulation is set to 300 nanoseconds.

The numerical experiments here utilize simulated ToA data for which the claimed

locations for legitimate and malicious vehicles in equal proportion are generated randomly in the specified area. The ToA from the claimed locations at the four BSs is calculated using equation (3.4). The receivers in the BSs are under the influence of independent thermal noise z_i, and thus the ToA measurements they make have a certain degree of variation. This variation (in nanoseconds) is extracted from a Gaussian random function with a fixed standard deviation. The area around the vehicles have blockings, and therefore their transmitted signals cannot reach the BSs directly; hence their ToAs have additional NLoS bias ϕ_i in them. To mimic reality, ϕ_i is extracted from an exponential distribution as given below

$$f(\phi_i) = \rho_i e^{-\rho_i \phi_i}, \tag{3.16}$$

where ρ_i is the scale parameter.

For the information-theoretic LVS, the Total Error, the false positive rate, and the detection rate are determined using equations (3.13) and (3.15). The data considered for the information-theoretic LVS analysis is used to also train the NN-LVS. This data is known as the training data.[1] In the training phase, the NN-LVS is fed with random vehicles data at a speed of one vehicle data per second. During each second, the NN-LVS is trained with the available training data. The backpropagation algorithm has a set of internal parameters to terminate the training phase for the NN-LVS. In most cases, the 'maximum validation failures', which is the maximum number of iterations in a row during which the NN-LVS's performance fails to improve or remains the same, terminates the training phase. The value for this parameter is set to 6. The weights and biases are considered optimized once the training phase has concluded. The NN-LVS afterwards can be used to classify a vehicle as legitimate or malicious in the test data[2].

The NN-LVS is trained during the 1st second with input training data from a

[1]The training data refers to the ToA data received from vehicles that are known *a-priori* to be legitimate or malicious. Use of such data in order to set the NN parameters, prior to its use on 'unlabeled' data (*i.e.*, data from vehicles which are not known *a-priori* to be legitimate or malicious), is known as the training phase.

[2]The test data is simulated under a different realization with the same settings as training data. Further, the test data has no labels.

single random vehicle. At the end of 1st second, the NN-LVS (with its weights and biases optimized) is subject to calculate a Total Error for the test data. In the 2nd second, another random vehicle training data is added to the previously available single-vehicle training data. The combined data forms a new training dataset which is used to retrain the NN-LVS (from 1st second). After retraining, the NN-LVS is used to determine a new Total Error for the test data. Yet another random vehicle training data is added to the previously available training data in the 3rd second, and the updated dataset is utilized to once again train the NN-LVS. At the end of the third second, a revised Total Error is calculated for the test data. This process of updating the training dataset, retraining the NN-LVS and recalculating a new Total Error for the test data continues in the following seconds. The Total Error keeps on decreasing with the passage of time.

In Fig. 3.2, the Total Error is initially determined for a dataset that has legitimate and malicious vehicles in equal proportions (i.e., P_0, and $P_1 = 0.5$). The standard deviation for z_i (extracted from a Gaussian distribution) is 300 nanoseconds, while the standard deviation for NLoS bias (extracted from an exponential distribution) is indicated by the different curves. The number of BSs used is 4. The LRT (*i.e.*, the Total Error arising from the information-theoretic LVS) corresponding to each NLoS curve is indicated by the dashed arrow lines. We can see that performance for the information-theoretic LVS deteriorates as the NLoS bias increases. In contrast, the performance for the NN-LVS improves with an increase in the NLoS bias in the ToA data. It is clear that the NN-LVS can accommodate the NLoS conditions significantly better than an information-theoretic LVS.

Next, the impact of changing P_1 (the proportion of malicious vehicles) is studied on the performance of the LVS. An NN-LVS is trained through similar procedures as described earlier, but the proportion of malicious vehicles in the test data varies. In one of the experiments, the standard deviations for z_i and the NLoS bias are fixed to 300 nanoseconds. The number of BSs is 4. In Fig. 3.3, we observe that the NN-LVS performs consistently even when P_1 is different in the test data. We can see that the NN-LVS performance is satisfactory even when the test data has 99.95% legitimate vehicles and 0.05% malicious vehicles. The red line in Fig. 3.3 shows the Total Error

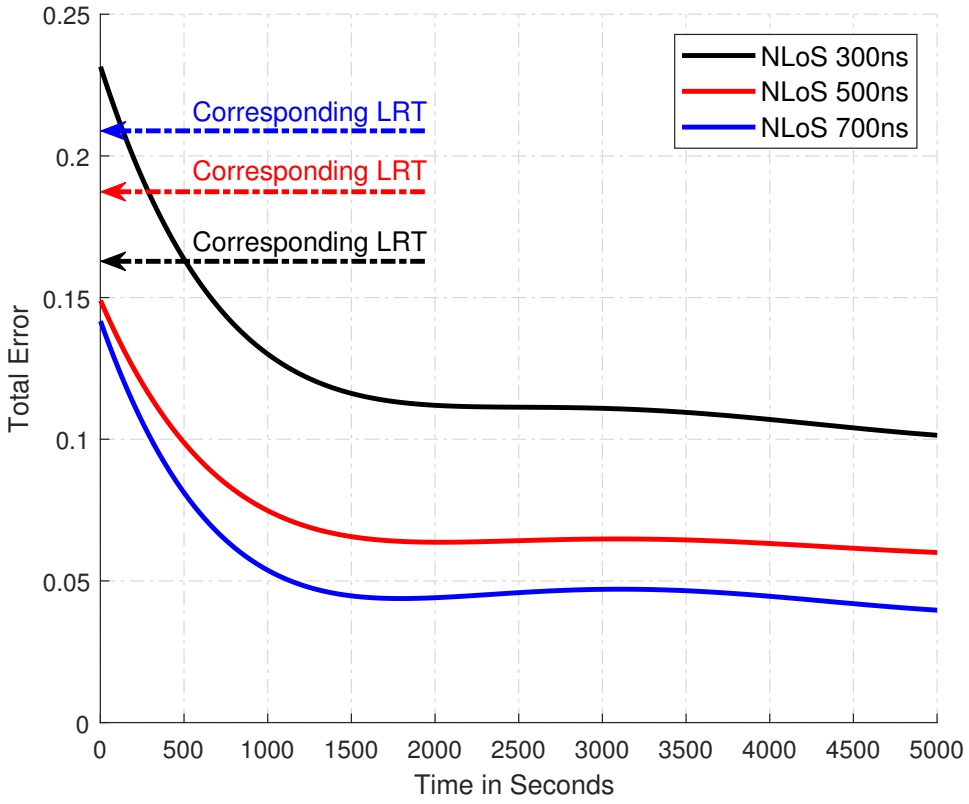

Figure 3.2: The Total Error performance of the NN-LVS with 4 BSs as it is being trained under changing NLoS bias conditions. The training and the testing data has legitimate and malicious vehicles in equal proportions. The variable z_i (extracted from a Gaussian random distribution) has a fixed standard deviation of 300 nanoseconds. The NLoS bias (extracted from an exponential distribution) has a varying standard deviation indicated by the different colours of the curves. The NN-LVS indicates an improved performance with a Total Error of 0.10 (NLoS 300ns), 0.06 (NLoS 500ns) and 0.04 (NLoS 700ns) as compared to the LRT method of [23], which gives a Total Error of 0.16 (NLoS 300ns), 0.18 (NLoS 500ns) and 0.21 (NLoS 700ns). Higher NLoS leads to easier discrimination between a legitimate vehicle and a malicious one placed far from the BSs.

for the information-theoretic LVS when the legitimate and malicious vehicles are in equal proportions in the data (with no changes to z_i and the NLoS bias). This study shows that, unlike the information-theoretic LVS, whose performance is conditioned on the *a-priori* knowledge of P_1, the NN-LVS's performance is largely independent of P_1.

In Fig. 3.4, the standard deviation for the NLoS bias is changed to 500 nanoseconds. The variable z_i is still extracted from a Gaussian random distribution with a standard deviation of 300 nanoseconds. We can observe that NN-LVS's performance

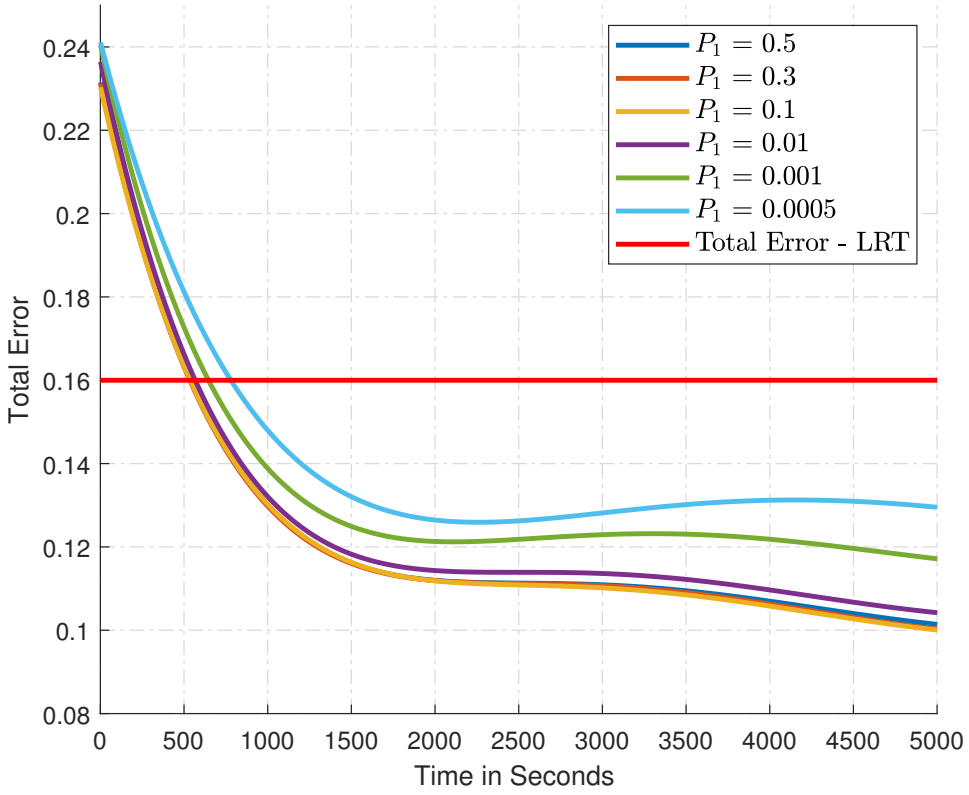

Figure 3.3: The Total Error performance of the NN-LVS with 4 BSs. The test data has different proportions for legitimate and malicious vehicles as highlighted by the different colour of curves. The variable z_i is extracted from a Gaussian random distribution with a standard deviation of 300 nanoseconds. The NLoS bias is extracted from an exponential distribution with a fixed standard deviation of 300 nanoseconds. The red line shows the Total Error for the information-theoretic LVS (based on LRT method) for a data (realized under same settings of standard deviation for z_i and NLoS bias) which has both P_0, and P_1 equal to 0.5. We can see that the NN-LVS performs consistently with different P_1 values in the test data.

is independent of the P_1 value.

In Fig. 3.5, the number of BSs is modified to 6 while the standard deviations for the NLoS bias and z_i are kept the same as those in Fig. 3.4. Still, we can see a promising performance from the NN-LVS under different P_1 values.

The NN-LVS framework proposed in this chapter can be applied to many sub-applications within IoT and 5G whose performance largely depends on the true and verified locations of users. Map services, smart parking, mMIMO, enhanced beamforming, coverage enrichment, and interference mitigation are just a few of the

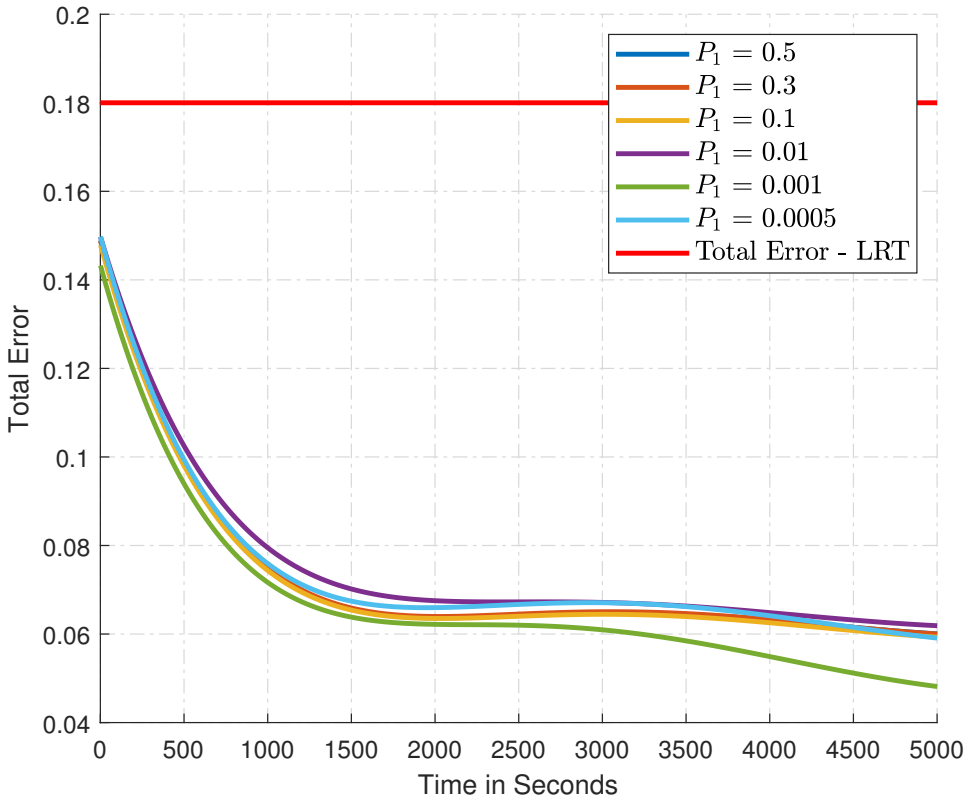

Figure 3.4: The Total Error performance of the NN-LVS as in Fig. 3.3 except the standard deviation for NLoS bias now is 500 nanoseconds.

applications which can benefit from an NN-LVS. The same framework can also assist in defence-related operations where location and its verification is of mission-critical.

3.5 Conclusion

Information-theoretic LVS frameworks, due to their operating limitations, are not practical in many real-world scenarios. To close this gap, the use of an NN approach to location verification has been proposed. This approach is particularly useful considering one of the key inputs to any LVS, *i.e.*, the knowledge on the proportion of vehicles anticipated to be malicious - an input usually unknown. Using simulated ToA data, it has been shown how an NN-LVS outperforms a state-of-the-art information-theoretic LVS. Unlike the information-theoretic LVS, the working of the NN-LVS is shown to be largely independent of the proportion of malicious vehicles

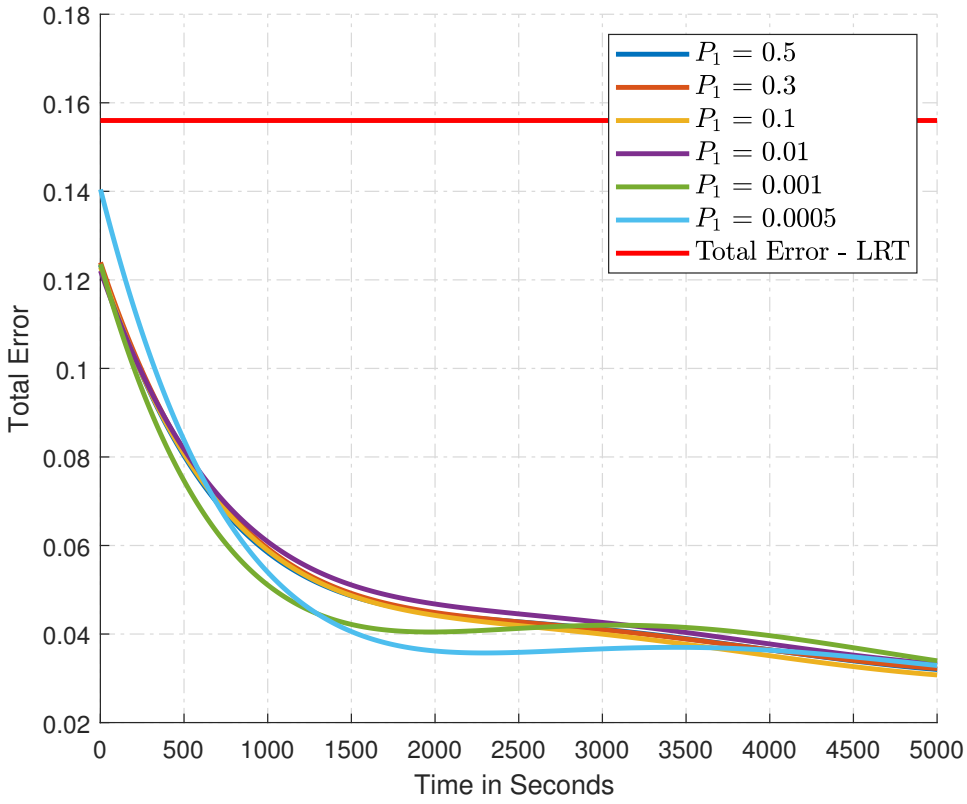

Figure 3.5: The Total Error performance of the NN-LVS as in Fig. 3.4 except the number of BSs now is 6.

in the area. Unknown channel conditions, such as NLoS bias effects, have also been shown to be better accommodated by the NN-LVS. The novel approach for enhancing the performance of real-world LVSs, formulated in this chapter, is believed to potentially form the foundation for all future works in this important area. In the next chapter, it will be demonstrated that NN-based LVSs are practical in real-world situations.

Chapter 4

Neural Network-based LVS from a Real-World Perspective

The claims made in the previous chapter were based on simulated ToA of the vehicles' transmitted signals. It was shown in the same chapter that the NN-LVS could outperform the traditional LVS when the simulated ToA of the vehicles' transmitted signals had additional biases in them. A factor causing such a bias was the internal thermal noise of the receiver. Additionally, the static and non-static blockings in the area also caused biases to the ToA of the vehicles' transmitted signals. The static blockings were due to the buildings in the area, while the non-static blockings were due to the temporary structures that existed in between the vehicles and the receiver (*i.e.*, other vehicles, etc.). Another important aspect studied in the previous chapter was the efficient performance of the NN-LVS in the absence of information on the proportions of genuine and malicious vehicles in the area. Such information is a prerequisite for working the information-theoretic LVSs (as was shown in the previous chapter).

A summary of the differences between the previous and current chapters is stated below

- Usually, claims made for an NN-LVS based on simulated data (as in the previous chapter) are unrealistic unless proven otherwise. This can be proven by

considering a practical scenario (as the case in this chapter) where experimental data is taken into account. In this chapter, for the first time, it is shown through real-world data that an NN-LVS is a more practical solution.

- Unlike Chapter 3, where ToA of the vehicles' transmitted signals is considered, this chapter takes into account a different physical layer property, *i.e.*, RSS of the vehicles' transmitted signals in the field. The RSS is preferred over the ToA in this chapter as such measurements can be quickly and more efficiently collected in real-world scenarios. Unlike the ToA, where (due to the sensitive nature of the measurements) sophisticated devices are required to collect the measurements, RSS measurements do not usually demand high calibrated and state-of-the-art tools.

- The narrative established in the Chapter 3 is entirely different from this chapter. This chapter accounts for the more efficient performance of the NN-LVS in scenarios where a malicious vehicle randomly claims its location away from its actual location. In addition, the NN-LVS also outperforms an information-theoretic LVS in scenarios where the malicious vehicle, to lessen its chances of being detected, utilize all its resources to optimize its attack location.

4.1 Introduction

VANETs utilize communications between vehicles and other infrastructure to assist with various traffic problems. VANETs can function based on V2V and/or vehicle-to-RSU communication [106]. RSUs are fixed BSs installed at specific locations with an aim to assist VANETs with their operations. An RSU (or a trusted vehicle whose location is *a priori* verified), can also function as a PC. The PC processes the communication data before issuing instructions to the vehicles under its coverage area.

As was highlighted in the previous chapter that vehicles claimed locations could have errors in them for multiple reasons, *i.e.*, software/hardware issues, extreme weather situations, or for launching location spoofing attacks. If not timely ad-

dressed, such location errors can result in a host of performance issues for VANETs. Furthermore, in extreme cases, a lack of position verification may lead to catastrophic situations such as vehicle collisions. Several location verification protocols have been devised to validate a vehicle's supplied location information [22, 24, 26, 27, 70, 72, 74–76, 81]. Almost all these protocols have a serious limitation in their operation - they normally operate efficiently if the channel conditions assumed at the time of their design [22] become a reality. However, such assumptions are far from reality and make the deployment of these protocols impractical in real-world conditions. NNs have been integrated into the location verification protocols to overcome such limitations. In a few such works [28, 84], the authors, through simulated data, have shown NN-based LVS protocols to be a step closer to reality. It has been demonstrated, through simulated data, in the previous chapter, how an NN-based solution for location verification has been able to operate efficiently in the absence of the assumptions that are required for the operation of the non-NN-based LVSs. What remains to be determined is whether these advances hold up under conditions where real-world data is input to the NN-LVS. This chapter, which represents the first experimental deployment of any NN-LVS, answers this question in the affirmative. The main contributions in this chapter are summarized below.

- For the first time, an NN-LVS analysis based on real-world data, namely RSS measurements, is carried out.
- It is shown that the NN-LVS outperforms an information-theoretic LVS when a malicious vehicle sets its claimed (untrue) location at some random location.
- It is also shown that, unlike the information-theoretic LVS, the NN-LVS still performs efficiently even when the malicious vehicle formally minimizes spoofing detection by optimizing its claimed (untrue) location.

The remainder of this chapter is organized as follows. Section 4.2 details the system model. Section 4.3 presents the performance analysis using information theory and NN techniques. Section 4.4 discusses the real-world experimental data collection. Section 4.5 provides numerical results, and Section 4.6 concludes the chapter.

4.2 System Model

The system model in this chapter is slightly different from the one in the last chapter. Unlike the previous chapter, where ToA measurements are utilized, this chapter considers RSS measurements of the vehicles transmitted signals in the context of VANETs. For convenience, the system model is stated again below:

1. The true location of a legitimate or malicious vehicle is denoted by $\mathbf{x}_t = [\mathrm{x}_t, \mathrm{y}_t]$.

2. The reported location from a legitimate or malicious vehicle is referred to as the 'claimed location' and is denoted by $\mathbf{x}_c = [\mathrm{x}_c, \mathrm{y}_c]$. The claimed location for a legitimate vehicle is exactly the same as its true location. On the other hand, a malicious vehicle spoofs its location, *i.e.*, its claimed location is not the same as its actual location.

3. For a malicious vehicle $||\mathbf{x}_c - \mathbf{x}_t|| \geq r$, where r is an *a priori* distance representing the minimum distance between its claimed and true locations.

4. The framework comprises of N RSUs as verifying BSs, with publicly known true locations. All RSUs are in the transmission range of the vehicles (whose claimed locations have to be verified). The true location of the i-th RSU is $\mathbf{x}_i = [\mathrm{x}_i, \mathrm{y}_i]$ where $i = 1, 2, 3, \ldots, N$.

5. One of the RSUs is chosen as PC. The PC accumulates its own RSS measurements with those collected by other RSUs for further processing. Finally, the PC decides on the integrity of a vehicle's claimed location.

6. Under the null hypothesis $\mathcal{H}_o$, the vehicle is considered as legitimate, *i.e.*, $\mathcal{H}_o : \mathbf{x}_c = \mathbf{x}_t$.

7. Under the alternative hypothesis $\mathcal{H}_1$, the vehicle is considered as malicious, *i.e.*, $\mathcal{H}_1 : \mathbf{x}_c \neq \mathbf{x}_t$.

Based on a log-normal path loss model, under $\mathcal{H}_o$, the RSS (all RSS in dBm) measured by the i-th RSU from a legitimate vehicle, y_i, is given by

$$y_i = u_i + w_i, \qquad i = 1,\ 2,\ 3, \ldots, N, \tag{4.1}$$

where w_i is a zero mean normal random variable with variance σ_T^2 representing the channel noise, and u_i is the mean RSS at i-th RSU. This latter quantity is given by

$$u_i = p_{d_o} - 10\,\gamma\,\log_{10}\left(\frac{d_i^c}{d_o}\right), \tag{4.2}$$

where p_{d_o} is a reference RSS at a reference distance d_o, γ is the path loss exponent, and d_i^c is the distance of a legitimate vehicle's claimed location to the i-th RSU, given by

$$d_i^c = \sqrt{(\mathrm{x}_c - \mathrm{x}_i)^2 + (\mathrm{y}_c - \mathrm{y}_i)^2}. \tag{4.3}$$

The measurements made by the N RSUs are independent of each other. Under $\mathcal{H}_o$, they collectively form a vector $\mathbf{y} = [y_1,\ y_2,\ y_3, \ldots, y_N]^\top$. Based on Eq. (4.1) the vector $\mathbf{y}$ follows a multi-variate normal distribution given as

$$f(\mathbf{y}|\mathcal{H}_o) \sim \mathcal{N}(\mathbf{u}, \Sigma), \tag{4.4}$$

where $\mathbf{u} = [u_1,\ u_2,\ u_3, \ldots, u_N]^\top$ is the mean RSS vector under $\mathcal{H}_o$, and $\Sigma = \sigma_T^2 \mathbf{I}_N$ is the covariance matrix with $\mathbf{I}$ as the identity matrix.

Under $\mathcal{H}_1$, a malicious vehicle spoofs its claimed location. It reports its claimed location to be at a minimum distance r (meters) away from its true location. As an example scenario - we can think of the malicious vehicle pretending to be on the road while actually placed in a nearby street. The RSS value measured by the i-th RSU from a malicious vehicle, y_i, is given by

$$y_i = v_i + w_i, \qquad i = 1,\ 2,\ 3, \ldots, N, \tag{4.5}$$

where v_i is given by

$$v_i = p_{d_o} - 10\,\gamma\,\log_{10}\left(\frac{d_i^t}{d_o}\right), \tag{4.6}$$

and d_i^t is the distance of its true location to the i-th RSU, given by

$$d_i^t = \sqrt{(\mathrm{x}_t - \mathrm{x}_i)^2 + (\mathrm{y}_t - \mathrm{y}_i)^2}. \tag{4.7}$$

The measurements made by the N RSUs are independent of each other. Under $\mathcal{H}_1$, they collectively form a vector $\mathbf{y} = [y_1,\ y_2,\ y_3, \ldots, y_N]^\top$. From Eq.(4.5), vector $\mathbf{y}$ follows a multi-variate normal distribution given as

$$f(\mathbf{y}|\mathcal{H}_1) \sim \mathcal{N}(\mathbf{v}, \Sigma), \tag{4.8}$$

where $\mathbf{v} = [v_1,\ v_2,\ v_3, \ldots, v_N]^\top$ is the mean RSS vector under $\mathcal{H}_1$.

4.3 Performance Analysis

The LVS produces a binary result for a vehicle, *i.e.*, legitimate or malicious. The performance of the LVS is measured using two methodologies; using information theoretic analysis as in [23], and through the newly designed NN-LVS method that makes use of NNs. In both cases, a Bayes average cost function is chosen as the performance metric for LVS in terms of 'Total Error'. The Total Error is given by

$$\xi = p(\mathcal{H}_o)\alpha + p(\mathcal{H}_1)(1 - \beta), \tag{4.9}$$

where $p(\mathcal{H}_o)$ and $p(\mathcal{H}_1)$ are the *a priori* probabilities of occurrences of $\mathcal{H}_o$ (*i.e.*, legitimate vehicle) and $\mathcal{H}_1$ (*i.e.*, malicious vehicle), respectively. Equal proportions of legitimate and malicious vehicles in the field are assumed, so both $p(\mathcal{H}_o)$ and $p(\mathcal{H}_1)$ are equal to 0.5 in this chapter. The variables α, and β represent the false positive rate, and the detection rate, respectively. The Eq. (4.9), therefore, can be

reformulated as

$$\xi = 0.5\alpha + 0.5\,(1-\beta)\,. \tag{4.10}$$

4.3.1 Information-theoretic LVS

From now on, the information-theoretic analysis in this chapter will be referred to as the LRT method. The LRT method requires some parameters and channel information to be available in advance. This information includes the path loss exponent γ, the mean RSS vectors highlighted in the system model, and the LRT decision threshold λ. It is well known that the LRT method achieves the optimum detection results for a given false positive rate [105]. This leads to the conclusion that the LRT minimizes the Total Error and maximizes the mutual information between input and output of the LVS [79]. The following decision rule is taken into account for the LRT method

$$\Lambda\left(\mathbf{y}\right) \triangleq \frac{p(\mathbf{y}|\mathcal{H}_1)}{p(\mathbf{y}|\mathcal{H}_o)} \underset{\mathcal{D}_0}{\overset{\mathcal{D}_1}{\gtrless}} \lambda, \tag{4.11}$$

where $\Lambda\left(\mathbf{y}\right)$ is the likelihood ratio, and $\mathcal{D}_1$ and $\mathcal{D}_0$ are the binary decision values (*i.e.*, whether the vehicle is malicious or legitimate), while $p(\mathbf{y}|\mathcal{H}_o)$ and $p(\mathbf{y}|\mathcal{H}_1)$ are given by

$$p(\mathbf{y}|\mathcal{H}_o) = \frac{1}{\sqrt[k]{2\pi}\sqrt{|\Sigma|}} e^{-\frac{1}{2}(\mathbf{y}-\mathbf{u})\Sigma^{-1}(\mathbf{y}-\mathbf{u})}, \tag{4.12}$$

$$p(\mathbf{y}|\mathcal{H}_1) = \frac{1}{\sqrt[k]{2\pi}\sqrt{|\Sigma|}} e^{-\frac{1}{2}(\mathbf{y}-\mathbf{v})\Sigma^{-1}(\mathbf{y}-\mathbf{v})}, \tag{4.13}$$

where $|\Sigma|$ is determinant of Σ. The decision rule given in Eq. (4.11) can be reformulated as

$$\Lambda\left(\mathbf{y}\right) \triangleq \frac{e^{-\frac{1}{2}(\mathbf{y}-\mathbf{v})\Sigma^{-1}(\mathbf{y}-\mathbf{v})}}{e^{-\frac{1}{2}(\mathbf{y}-\mathbf{u})\Sigma^{-1}(\mathbf{y}-\mathbf{u})}} \underset{\mathcal{D}_0}{\overset{\mathcal{D}_1}{\gtrless}} \lambda. \tag{4.14}$$

With equal proportions of legitimate and malicious vehicles, the variable λ equals 1. The malicious vehicle is assumed to be optimizing its claimed location. That is, through an optimization strategy, it minimizes its probability of being detected by the LVS. It is assumed in this chapter that the malicious vehicle's optimum claimed location is constrained to be within the transmission range of the RSUs. To optimize its claimed location under such a constraint and to reduce its chances of being detected (or we can say the detection rate), the malicious vehicle attempts to minimize the statistical distance between the two probability distributions, *i.e.*, $f(\mathbf{y}|\mathcal{H}_1)$ and $f(\mathbf{y}|\mathcal{H}_o)$. The malicious vehicle can take into account any of the available statistical optimization strategies to minimize this statistical distance. Here, the malicious vehicle adopts the famous KL divergence strategy for this purpose. The KL divergence can be used for both discrete and continuous probability distributions. In the latter case, the integral of the two probability distributions is calculated instead of the sum of the probabilities of the discrete events. Since, the probability distributions are continuous in our case, the malicious vehicle minimizes the KL divergence between $f(\mathbf{y}|\mathcal{H}_1)$ and $f(\mathbf{y}|\mathcal{H}_o)$ as below [24, 107, 108]

$$\begin{aligned} D_{KL}(f(\mathbf{y}|\mathcal{H}_1)||f(\mathbf{y}|\mathcal{H}_0)) &= \int_{-\infty}^{\infty} f(\mathbf{y}|\mathcal{H}_1) \ln \frac{f(\mathbf{y}|\mathcal{H}_1)}{f(\mathbf{y}|\mathcal{H}_0)} d\mathbf{y}, \\ &= \frac{1}{2}(\mathbf{v}-\mathbf{u})^T \boldsymbol{\Sigma}^{-1}(\mathbf{v}-\mathbf{u}). \end{aligned} \tag{4.15}$$

Then, the optimal claimed location $\mathbf{x}_c^*$ for the malicious vehicle can be obtained through

$$\mathbf{x}_c^* = \underset{||\mathbf{x}_t-\mathbf{x}_c||\geq r}{argmin}\ D_{KL}(f(\mathbf{y}|\mathcal{H}_1)||f(\mathbf{y}|\mathcal{H}_0)). \tag{4.16}$$

4.3.2 Neural Network-based LVS

This section highlights the novel approach used to design a classification framework for the verification of a vehicle's claimed location through supervised NN techniques. Feedforward NNs are well known for their performance in classification problems. A multi-layer feedforward neural network is utilized for the binary classification of a vehicle as either legitimate or malicious.

Based on the recommendations in Chapter 2, an NN-LVS framework with the raw inputs in its input layer, a 10-neuron hidden layer, and a 1-neuron binary output layer is finalized. The framework considers $\mathbf{y}$ (the RSS observation vector measured in the field) and the vehicle's claimed location as inputs. Several experiments are also conducted with different transfer functions in various layers of the NN-LVS. However, the results shown in the next section adopt the hyperbolic tangent-sigmoid transfer function in the hidden layer and the linear transfer function in the output layer. The NN-LVS utilizes the Levenberg-Marquardt as its backpropagation algorithm.

4.4 Real-world RSS Data Collection

RSS measurements from the vehicles were collected in a 150 X 150 meters area by 3 RSUs (an area that mimics a broad cross-section of two highways). Three devices were used to mimic three RSUs. The RSUs independently measure the RSS from the vehicles in the field at a rate of one RSS measurement per second per RSU simultaneously. The origin of the area is set to the location of RSU-1, as shown in Fig. 4.1. Moving Wi-Fi modems (802.11g) equipped with single antennas (at the same height as RSUs antennas) were transmitting at 2.437 MHz. Each Wi-Fi modem had an attached GPS unit. The attached GPS units (used to record the vehicles' locations every second) were used to mimic slow-moving vehicles. The GPS locations of these 'vehicles' were reported to the RSUs every second. The RSS measurements by individual RSUs and the vehicles' GPS locations were combined with the help of timestamps (available with both the measured RSS and the vehicles' GPS locations). The collected data (*i.e.*, the RSS measurements and the vehicles' GPS locations) were plotted to investigate any outliers. Following standard practices and a detailed analysis, the outliers were removed in the collected data. The refined data was standardized before utilizing it for further studies.

The path loss exponent γ is required for the LRT, and is determined directly from the field measurements via a linear fit of the measured RSS values against the logarithm of the distance to an RSU. The vectors $\mathbf{u}$ and $\mathbf{v}$ are calculated using (4.2)

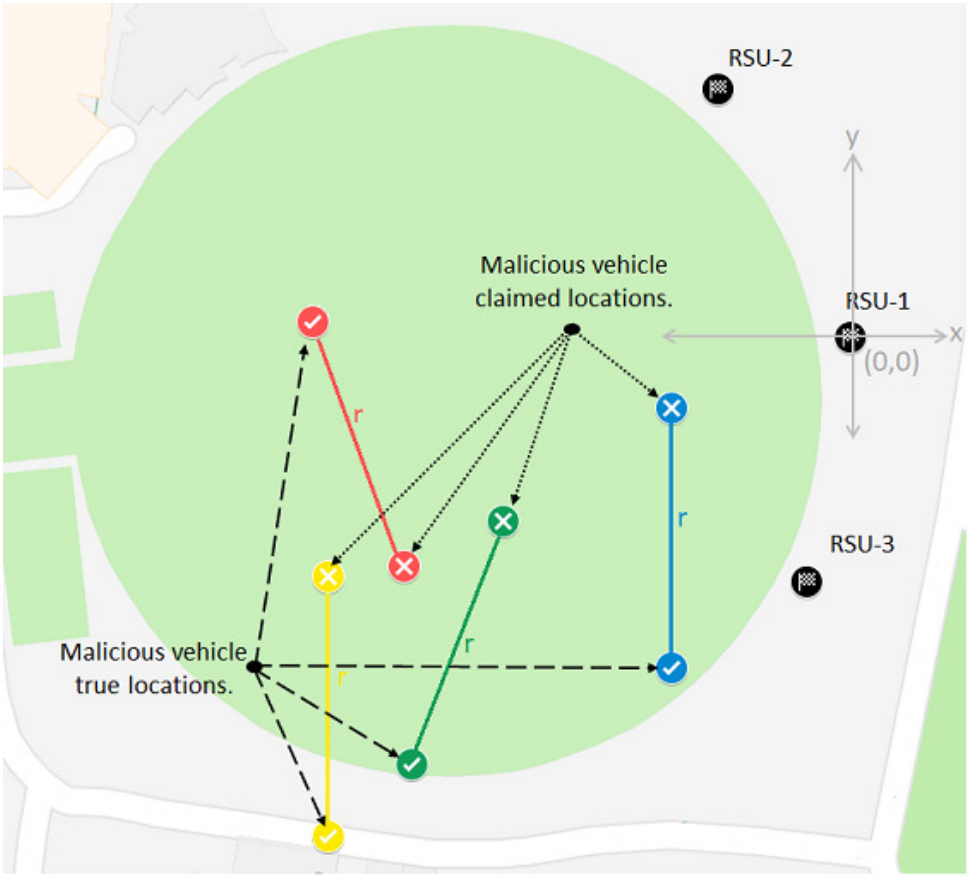

Figure 4.1: Malicious vehicles fake their locations to launch a location-spoofing attack. They report their claimed locations r meters away from their respective true locations. This figure only shows a sample of the malicious vehicles' true locations, their simulated random claimed locations at r meters, and the true locations for the RSUs. The value of r in this figure is 50 meters.

and (4.6) under the corresponding hypothesis. The variable σ_T is calculated using the mean RSS vector and the RSS measurements (made by each RSU).

The RSS measurements data is randomized and equally divided into two halves, with one half representing the legitimate vehicles and the other half representing the malicious vehicles. To launch a location-spoofing attack, the malicious vehicles spoof their locations by a minimum distance of r meters away from their true locations. Random claimed locations for the malicious vehicles are simulated by taking into account the distance constraint r. Fig. 4.1 highlights true and simulated random claimed locations for a sample of the malicious vehicles.

4.5 Numerical Results

Numerical results based on analysis from the LRT method and NN-LVS are presented next. In Fig. 4.2, it is assumed that the malicious vehicles *randomly* forge their claimed locations at a minimum distance of r meters from their true locations, but the claimed locations are within the transmission range of the RSUs. The Total Error is plotted against the number of training data used. For the LRT-based LVS, the Total Error, the false positive rate, and the detection rate under different values of r are calculated using Eqs. (4.10) and (4.14). The Total Error for r equal to 100m, 75m and 50m, is 0.05, 0.22, and 0.29, respectively (different colored-dashed arrows).

The data considered for the LRT-based LVS in Fig. 4.2 is also considered for the NN-LVS. Unlike the LRT method, where the LVS requires *a priori* information for the channel parameters, the NN-LVS only uses the measured RSS (at the RSUs) and the vehicles' reported claimed locations. The performance of the NN-LVS is independent of the absolute location of the vehicles and RSUs. This data, which has legitimate and malicious vehicles in equal proportions, is randomized and divided into two data sets; a training set with 80% of the entire data and a test set with the remaining 20% of the data. The training set also has data labels (legitimate or malicious). These data labels indicate whether a particular training sample represents a legitimate or malicious vehicle. The use of such data is required to set the weights and biases for the NN-LVS in the training phase. On the other hand, the data in the test set has no such labels, which means that we have no *a priori* information if a particular sample belongs to a legitimate or a malicious vehicle. Once trained, the NN-LVS can be used to classify vehicles in the test set.

In the training phase in Fig. 4.2, the NN-LVS is supplied with training samples from the training data at a rate of one random training sample per unit time and plot the Total Error for the test set after each unit time. The NN-LVS's backpropagation algorithm terminates the training phase once a threshold for any of its internally set parameters is met. It is observed that in most cases, the 'maximum validation failures' parameter of the backpropagation algorithm (the maximum number of se-

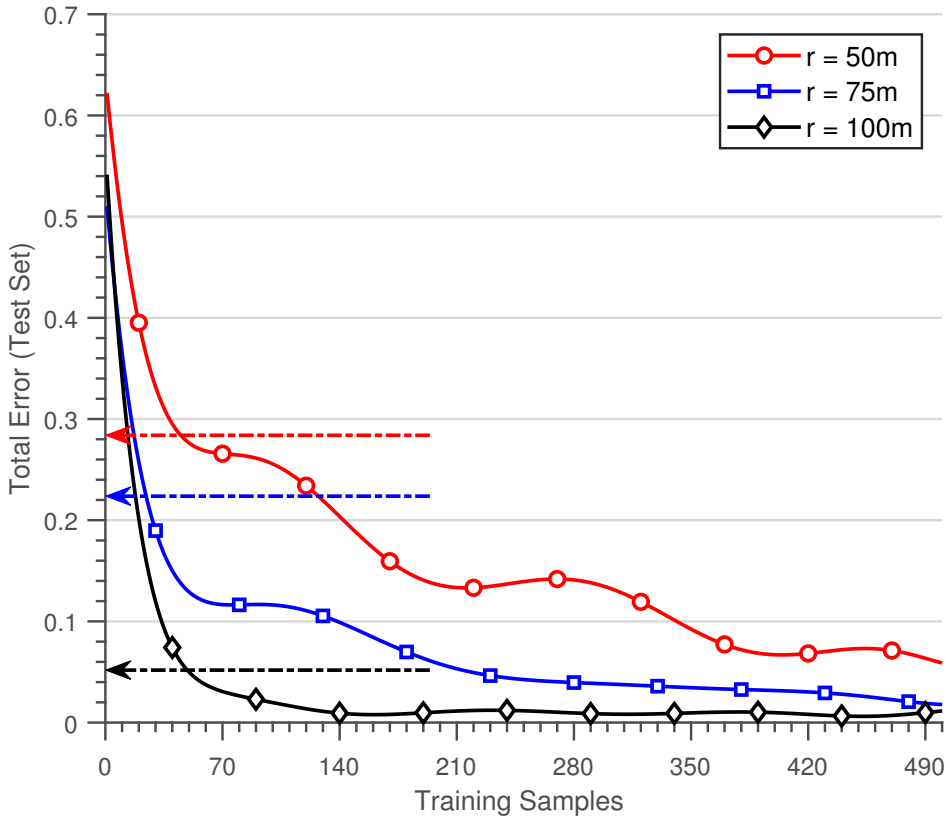

Figure 4.2: A comparison study where an NN-LVS outperforms an LRT-based LVS. The NN-LVS with no *a priori* channel information achieves a final Total Error (indicated by solid lines) of 0.01, 0.02, and 0.06, for r equal to 100m, 75m, and 50m, respectively, for the data in the test set. On the other hand, the LRT-based LVS with *a priori* channel information achieves a Total Error (indicated by dashed arrows) of 0.05, 0.22, and 0.29, for r 100m, 75m, and 50m, respectively, for the data in the test set. Note, in these calculations, the malicious vehicles do not optimize their claimed locations.

quential iterations in which the NN-LVS's performance fails to improve) is reached, and this terminates the training phase. A value of 6 is set for this parameter. This trained NN-LVS is then used to classify vehicles in the test set as either legitimate or malicious. This procedure is repeated for each sample of the training data shown in Fig. 4.2. As shown in Fig. 4.2, as expected, the Total Error for the test set improves as the training continues. The final Total Error for the test set (after 500 training samples) using the NN-LVS for r equal to 100m, 75m and 50m, is 0.01, 0.02, and 0.06, respectively. It is evident from Fig. 4.2 that the NN-LVS with no *a priori* channel information has much-improved performance relative to the LRT-based LVS.

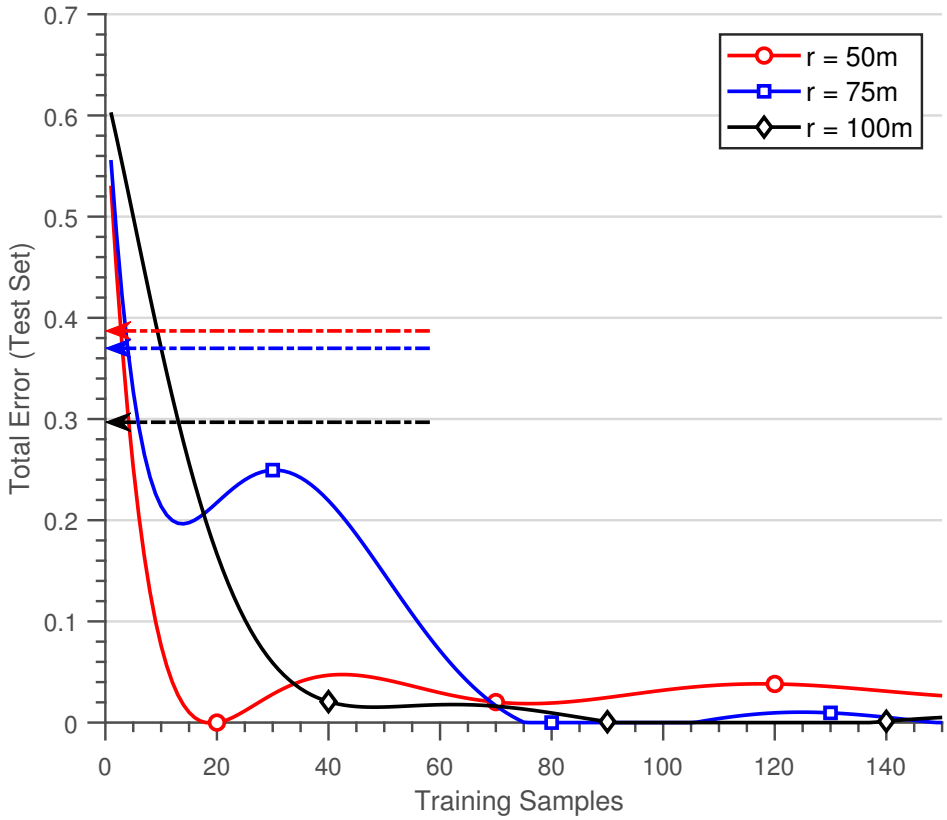

Figure 4.3: A comparison study of the LRT-based LVS with NN-LVS as in Fig. 4.2 except the malicious vehicles now optimize their claimed locations.

The malicious vehicles are now assumed to overhear the communication between the legitimate vehicles and the RSUs. The malicious vehicles use this information to best optimize their claimed locations ($\mathbf{x}_c = \mathbf{x}_c^*$) prior launching a location-spoofing attack. That is, they set their claimed location using Eq. (4.15) so as to minimize their probability of being marked malicious by the LVS.

In Fig. 4.3, the performances of the NN-LVS and the LRT-based LVS are compared. We see again that the NN-LVS still outperforms the LRT-based LVS. However, we notice a rather counter-intuitive finding where, compared to Fig. 4.2, the Total Error for the NN-LVS improves much faster. This counter-intuitive finding is as a result of the geometry of the RSUs in this specific experiment. This geometry leads to a clustering in the malicious vehicles' claimed location settings. In general (*i.e.*, more general RSU geometries), if the malicious vehicles' optimize their claimed locations, the Total Error for the NN-LVS is expected to take longer to reach its asymptotic value.

4.6 Conclusion

Information-theoretic LVS frameworks, due to their operating limitations, are not practical in many real-world scenarios. To address this gap, the use of an NN approach to location verification is proposed. This new approach is handy since, unlike an information-theoretic LVS, an NN-LVS does not require *a priori* information on the channel parameters. Additionally, the NN-LVS can adapt itself to any changing channel conditions.

Using real-world RSS data, it is shown how a deployed NN-LVS outperforms a state-of-the-art information-theoretic LVS. Further, it is demonstrated how this result holds even when the adversary optimizes its attack location. It is believed the novel approach for enhancing the performance of real-world LVSs that is developed here potentially forms the foundation for all future works in the important area of wireless location verification.

In the next chapter, a key area related to the performance of an NN-LVS, *i.e.*, the training time, is focussed. Detailed investigation on how the training time for an NN-LVS can be shrunk without impacting its performance is carried out.

Chapter 5

Optimization of the Training Time of a Neural Network-based LVS

The training time of an NN is a key parameter that defines the viability of a designed NN-based framework. This chapter considers the best aspects of information theory and NN concepts to provide a novel solution to optimize the NN-LVS's training time. A derived information-theoretic bound is utilized to limit the training time for the NN-LVS such that the overall location verification performance is not compromised. Considering the ToA measurements of the vehicles transmitted signals, the performance for the NN-LVS is validated in a variety of channel conditions with varying threat situations. It is concluded that the NN-LVS adopts to the changing channel requirements in a satisfactory way.

5.1 Introduction

Location information forms the basis of almost all network decisions in VANETs. Within most vehicular network architectures, a vehicle usually obtains its position from the GPS, perhaps with assistance from other components of the wider GNSS. The Wireless Access in Vehicular Environments (WAVE), specified in the IEEE 1609 family of standards [109], dictates that such location information is periodically re-

ported to the broader network. A possibility exists whereby a malicious vehicle, in order to get an advantage over other vehicles, or to disrupt the system, or to trigger an unavoidable collision, can deceive the wider network by forging its location. Alternately, there is a possibility that a GPS receiver may wrongly calculate a vehicle's location due to poor reception or other non-malicious circumstances. Such events, if not carefully considered, can result in a host of undesirable outcomes.

Due to such considerations, numerous algorithms for validating a vehicle's reported location in VANETs have been proposed over the past few years [22, 24, 26, 27, 73, 79, 81]. These algorithms, in general, make use of a network infrastructure that comprises RSUs and vehicles. RSUs are static BSs placed at optimum locations to assist VANETs with a range of communication objectives. The inter-vehicle, RSU-to-vehicle and vehicle-to-RSU communication data is processed by an RSU in a specified coverage area so at best facilitate network operations. However, in the context of location verification, these same signals can be utilized to verify all reported GPS positions.

Researchers have formulated a number of information-theoretic LVSs over the past few years [22] which utilize available signal metrics, such as the RSS of a transmitted signal from a vehicle, the ToA of the transmitted signal from the vehicle and/or the AoA of the transmitted signal from the vehicle, to validate the vehicle's reported location. At the same time, these LVSs have certain operation limitations in terms of the channel conditions and can only work under the assumptions made at the time of their initial design. Recently, NNs have added to many aspects of modern society through the use of its specialized algorithms. Enhanced clinical diagnosis [31], drug discovery [110], computer vision [111], speech recognition [112], efficient navigation [113], are a few of the areas impacted. NNs have also laid the foundation for autonomous vehicles [114]. The integration of an NN into LVS is crucial. Once trained to a specific limit, an NN-LVS can perform well in the channel conditions they are designed for [28, 97]. Due to their adaptability, the NN-LVS can accommodate the changing channel and environmental conditions, which ordinary LVSs lack. Further, the NN-LVS is believed to address multiple threat situations alone, thus eliminating the need to design threat-specific LVSs [21]. The NN-LVS

also has the potential to update itself through continuous learning, thus avoiding the risk of becoming obsolete. But, such an NN-LVS framework utilize the traditional stopping criteria to conclude its training. These traditional stopping criteria may unnecessarily stretch the training time of the NN-LVS, making its quick deployment difficult. This chapter addresses this limitation of the NN-LVS so as not to affect its overall location verification performance.

The novel contributions in this chapter can be described thus.

- The NN schemes that are developed for location verification deploy new techniques based on optimal-decision-theory frameworks. These techniques significantly reduce the NN-LVS training phase with only a minimal impact on performance.
- The threat model is studied under changing distance constraints for the malicious vehicle in an environment that closely relates to the real-world conditions. It is further shown how the designed NN-LVS beats the performance of an information-theoretic LVS under such constraints.
- A specific example of these conceptual NN frameworks is investigated by determining a new theoretical lower bound on the optimal information-theoretic performance of location verification within the context of NLoS effects as described by an exponential distribution of bias.
- By using this new lower bound, the trade-off in NN training time *vs.* performance is quantified, showing how traditional training times (using traditional stopping criteria) can be cut substantially while impacting on the Total Error only marginally.
- The performance of the NN-LVS is investigated under a wide range of generalized conditions in which the incoming data is substantially different from the data used to train the NN-LVS.

In this chapter, LVS deployment issues are first highlighted. This is followed by a brief review of the techniques that have been proposed to address the location ver-

ification problem within VANETs, discussing a range of formal location verification techniques that are based on optimal-decision theories. The usefulness of NNs for location verification is described next. Afterwards, the performance of the designed NN-LVS is compared to a state-of-the-art information-theoretic LVS in situations where the threat model for the malicious user is randomly changing. Finally, an analysis is carried out in view of the derived information-theoretic bound on the performance of an LVS in the context of biased timings in realistic channel conditions (Appendix C) before quantifying how the interplay between information theory and NN leads to improved location verification outcomes.

5.2 Related Works

5.2.1 LVS Deployment

In a configuration of a vehicular network where an LVS is deployed, several of the vehicles will have been already authenticated, but one vehicle (e.g., Vehicle A), has not. Vehicle A sends a request to join the network and reports its GPS position through its nearest RSU to the LVS server. All RSUs and nearby vehicles hearing this request report any relevant signal metrics (e.g., RSS) back to the server. The LVS server acts on this information, processing it to arrive at a verification decision. This decision is then reported back to all other vehicles in the network (or at least those in the vicinity of the requesting vehicle). In the case where the reported GPS position is not verified, the requesting vehicle has its security certificates revoked [109]. This communications system upon which this process is deployed over is quite generic and versatile. Dedicated short-range communications and the larger WAVE suite of protocols can easily accommodate the above configuration [109]. Vehicular network communications built on emerging 5G standards and beyond could also be easily accommodated.

5.2.2 Optimal-Decision Theory for Location Verification

The generic operational procedures of an LVS involve inputs that include the 'claimed location' of a user along with independent signal inputs such as RSS, ToA, and/or AoA. After processing these inputs, the LVS labels the user's reported (claimed) position to be either true or false. We note that an LVS solves a different problem than the determination of the location of a node in a secure fashion. In an LVS, we are focused on authenticating the validity of a claimed location reported by a node, and not on a node self-locating itself in a secure manner.

The following scenario is considered for a general LVS. A claimed location for a vehicle (usually a GPS position reported by that vehicle) is provided as input to the LVS processor along with the channel metrics (e.g., RSS, ToA, or AoA). The assumed channel conditions between the verifier and claimant vehicles will form part of the integrated system model adopted by the LVS. In general, it is assumed that the verifiers' locations are known and trustworthy. It could be that the verifiers are RSUs or other vehicles that have been previously authenticated. The LVS processor acts on these inputs with some pre-determined algorithm to produce a binary decision regarding the validity of the claimed location reported by the vehicle. A 'true' means that the reported location is genuine or trustworthy, while a 'false' means that the reported location has not passed the verification test and the corresponding vehicle is likely spoofing its location (or its onboard GPS is unreliable). This information can then be distributed to the wider network and it can be left up to individual vehicles, or the network system as a whole, on how to act on that information. In most situations, a vehicle failing the location verification test will have its security certificates revoked [109]. The nature of the specific algorithm deployed within the LVS can be wide and varied, ranging from simple heuristic algorithms [22, 58, 60, 61, 63–65] to those using optimal information-theoretic techniques, which will be discussed in the following.

Several location verification frameworks for VANETs, using optimal-decision theories, were formulated in a series of papers [26, 79, 81]. In [26], the use of directional antennas within the context of optimal decision theory and location verification for

vehicular networks was reported, showing how the use of such antennas can increase verification performance. In [79], the mutual information between the input and output LVS data was used as the objective optimization criterion. It was proven how a LRT, constructed from the probability of receiving a specific RSS value under different binary hypothesis (the claimant is truthful or untruthful), leads to an optimal decision. The work was extended in [81] to correlated-shadowing environments where a more general optimization criterion based on the Total Error was utilized. The Total Error is a combination of the fraction of false positives (friendly vehicle stating a true location marked as malicious) and the fraction of missed detections (malicious user spoofing a location marked as friendly). The use of optimal decision theory based on Total Error in the context of Rician channels (the most likely form of a channel in many vehicular environments) was reported in [81].

5.2.3 The Role of Neural Networks

The LVS processing discussed above will provide for the optimal decision conditioned on one important assumption - that the wider system model adopted is reality. However, in practice, the adopted system model can only be considered an approximation to reality. This is because inferred channel conditions, noise parameters, receiver characteristics, position-error assumptions, and other system descriptors are very difficult to *a priori* determine. The issue is further complicated by the fact that the system parameters are likely to possess both spatial and temporal characteristics. To accommodate such real-world issues, a more sophisticated solution based on NNs will be desired.

NNs within the context of vehicular networks have been discussed previously. It is considered that NNs will play an essential role in many characteristics of the vehicular networks, all the way through from vehicle classification problems, vehicle vision systems, autonomous driving, big-data analytics, routing, vehicle security, and driver behavior, to name just a few (see [38] and references therein for an overview). NNs have recently been used for the theoretical simulation of an in-region LVS [84]. It is believed that the performance for the NN-based LVSs in the context of IoT

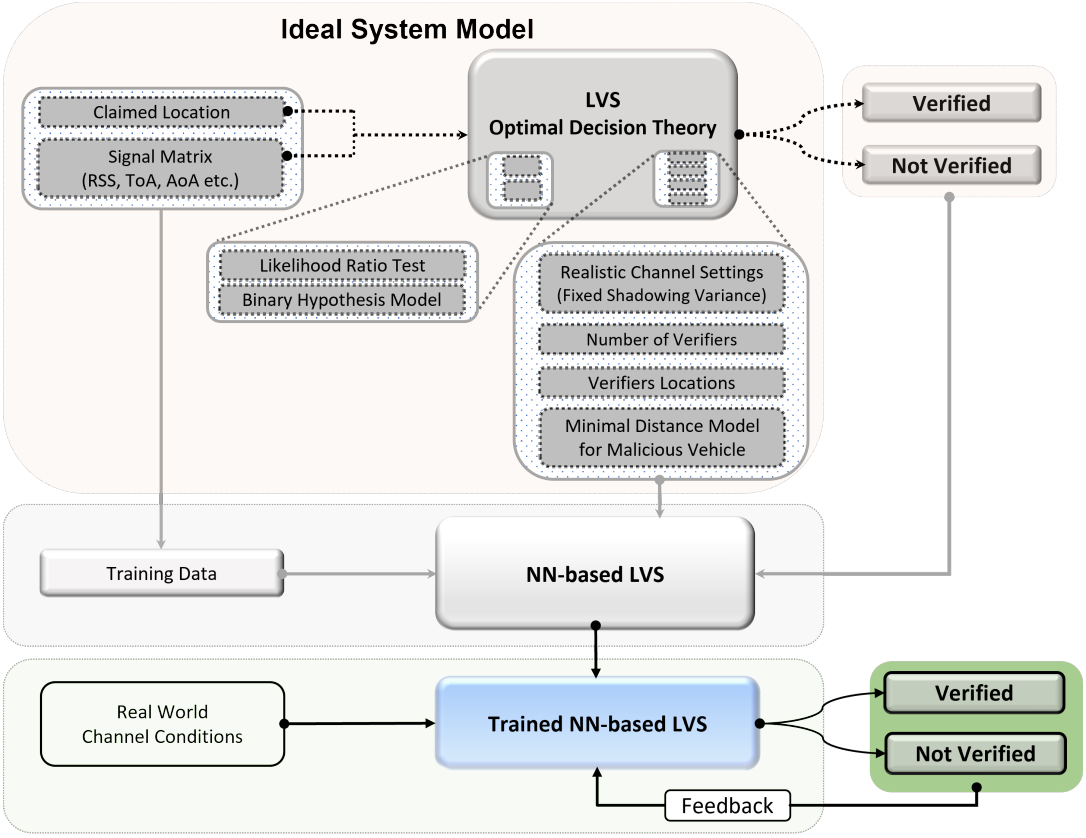

Figure 5.1: Inclusion of NNs into the LVS.

can be improved further by considering faster, extra reliable, and secure modern-communication frameworks [77, 115–119]. These new communication frameworks will secure and strengthen the transmission backbone for the NN-based LVSs.

5.3 System Model

5.3.1 The New LVS

It is believed the combination of the NN algorithms and information-theoretic techniques will provide for a 'best of both worlds' approaches to the critical problem of location verification. The optimal decision theory will be used as part of the initial training set for the NN and will be utilized in circumstances when the NN algorithm has not been trained for the specific locale being investigated. Once trained under ideal conditions, the NN algorithm within the architecture will then be adjusted

under real-world conditions when enough authenticated vehicles can be utilized as part of the training set.

A schematic of the combined framework architecture involving NNs is given in Fig. 5.1. Here we see how the optimal decision theory outputs can be used as part of the initial training phase, as well as the ongoing training phase (as more vehicles are authenticated or marked malicious). The mutual information between the outputs of the optimal-decision theory and the NN can be used as an additional input to the NN-base LVS, as well as providing a measure of 'agreement' between the two processes. Confidence levels associated with this measure could further assist the LVS.

5.3.2 Adopted System Model Assumptions

Building on the work of [23], the following system model is assumed:

1. A *single* vehicle to be identified (henceforth referred to as the user-vehicle) reports its claimed location, $\boldsymbol{\theta}_c = [\mathrm{x}_c, \mathrm{y}_c]$, to a network with N verifiers. Verifiers are fixed BSs with known locations. The location of the i-th BS is $\boldsymbol{\theta}_i = [\mathrm{x}_i, \mathrm{y}_i]$ $(i = 1, 2, 3, \ldots, N)$. One of the N BSs is selected as the PC - which collects all measurements.

2. A user-vehicle (genuine or malicious) obtains its true position, $\boldsymbol{\theta}_t = [\mathrm{x}_t, \mathrm{y}_t]$, from GPS with zero localization error. A genuine user-vehicle's claimed (reported) position, $\boldsymbol{\theta}_c$, is taken to be identical to its actual position $\boldsymbol{\theta}_t$. A malicious user-vehicle will falsify (spoof) its claimed position in an attempt to fool the LVS. The malicious user-vehicle's true location is unknown and its (spoofed) claimed location is taken to be $\boldsymbol{\theta}_c$.

3. The null hypothesis, where the user-vehicle is considered genuine, is denoted as $\mathcal{H}_0$. On the other hand, the alternative hypothesis, where the user-vehicle

is assumed malicious, is denoted as $\mathcal{H}_1$. This can be summarized as

$$\begin{cases} \mathcal{H}_0 : \ \boldsymbol{\theta_c} = \boldsymbol{\theta_t} \\ \mathcal{H}_1 : \ \boldsymbol{\theta_c} \neq \boldsymbol{\theta_t}. \end{cases} \tag{5.1}$$

5.3.3 Observation Model Under $\mathcal{H}_0$

The ToA measured at the i-th BS from a genuine user-vehicle, y_i, is given by

$$y_i = u_i + x_i + \phi_i, \quad i = 1, 2, 3, \ldots, N, \tag{5.2}$$

where

- $u_i = d_i^c/c$, with d_i^c as the Euclidean distance from the i-th BS to a user-vehicle's claimed location (also its true location) given by

$$d_i^c = \sqrt{(\mathrm{x}_c - \mathrm{x}_i)^2 + (\mathrm{y}_c - \mathrm{y}_i)^2}, \tag{5.3}$$

  and c as the speed of light,

- The variable x_i is a zero-mean normal random variable with variance σ_T^2, which represents the additive noise term in the ToA measurements, and

- ϕ_i is a random variable (in ns) that represents the bias caused by NLoS channel conditions, that follows an exponential distribution with ρ_i as the scale parameter, *i.e.*,

$$f(\phi_i) = \rho_i e^{-\rho_i \phi_i}. \tag{5.4}$$

In the first instance, the exponential distribution is adopted to model the bias term in the ToA measurements since this bias should always be positive due to the NLoS channel conditions.

The observations collected by different BSs are assumed to be independent and

thus, the likelihood function of the N-dimensional observation vector $\mathbf{y}$ under $\mathcal{H}_0$ is given by

$$f\left(\mathbf{y}|\mathcal{H}_0\right) = \prod_{i=1}^{N} f\left(y_i|\mathcal{H}_0\right), \tag{5.5}$$

where $f\left(y_i|\mathcal{H}_0\right)$ is the likelihood function of each y_i under $\mathcal{H}_0$. The explicit expression of $f\left(y_i|\mathcal{H}_0\right)$ is derived later to facilitate the determination of the likelihood function $f\left(\mathbf{y}|\mathcal{H}_0\right)$.

5.3.4 Observation Model Under $\mathcal{H}_1$

It is reasonable to assume that the malicious user-vehicle's true location is not close to its claimed location when an attack occurs. Note also, the malicious user-vehicle can alter some system parameters to interfere with the observations collected by all BSs. As such, the ToA measured at the i-th BS, y_i, is given by

$$y_i = T_x + w_i + x_i + \phi_i, \quad i = 1, 2, 3, \ldots, N, \tag{5.6}$$

where

- T_x is the time bias utilized by the malicious user-vehicle, and
- $w_i = d_i^t/c$, with d_i^t as the Euclidean distance from i-th BS to a user-vehicle's true location given by

$$d_i^t = \sqrt{(\mathrm{x}_t - \mathrm{x}_i)^2 + (\mathrm{y}_t - \mathrm{y}_i)^2}. \tag{5.7}$$

Since all BSs can communicate with each other, T_x is a constant value as seen by all BSs. For later convenience, $v_i = T_x + w_i$. Again, assuming that the observations collected from different BSs are independent, the likelihood function of the N-dimensional observation vector $\mathbf{y}$ under $\mathcal{H}_1$ is given by

$$f\left(\mathbf{y}|\mathcal{H}_1\right) = \prod_{i=1}^{N} f\left(y_i|\mathcal{H}_1\right), \tag{5.8}$$

where $f(y_i|\mathcal{H}_1)$ is the likelihood function of each y_i under $\mathcal{H}_1$. In general, the malicious user-vehicle will optimally set the value T_x in some sense to minimize the probability of being detected. The optimal value of T_x is denoted as T_x^*. It will be assumed in the calculations to follow that the malicious user-vehicle optimizes T_x through minimizing the KL-divergence between $p(\mathbf{y}|\mathcal{H}_0)$ and $p(\mathbf{y}|\mathcal{H}_1)$, where $p(\mathbf{y}|\mathcal{H}_0)$ and $p(\mathbf{y}|\mathcal{H}_1)$ are the likelihood functions of observation vectors $\mathbf{y}$ ($\mathbf{y} = y_1, y_2, y_3, \ldots, y_N$) under $\mathcal{H}_0$ and $\mathcal{H}_1$, respectively. This KL-divergence is defined as [23, 120]

$$\begin{aligned} D_{KL}(p(\mathbf{y}|\mathcal{H}_0)||p(\mathbf{y}|\mathcal{H}_1)) &= \int p(\mathbf{y}|\mathcal{H}_0) \ln \frac{p(\mathbf{y}|\mathcal{H}_0)}{p(\mathbf{y}|\mathcal{H}_1)} d\mathbf{y}, \\ &= \sum_{i=1}^{K} \frac{(u_i - v_i - T_x)^2}{2\sigma_T^2}. \end{aligned} \tag{5.9}$$

Then, the optimal value of T_x can be obtained through

$$\begin{aligned} T_x^* &= \arg \min_{T_x} D_{KL}\Big(p(\boldsymbol{m}|\mathcal{H}_0)||p(\boldsymbol{m}|\theta_t, \mathcal{H}_1)\Big), \\ &= \frac{1}{N} \sum_{i=1}^{N} (u_i - v_i), \end{aligned} \tag{5.10}$$

where $\boldsymbol{m}$ is the mean measurement vector for ToA of the signal under the relevant hypothesis. The value of T_x adopted by the malicious user-vehicle is not known to an LVS, and therefore the examination based on T_x^* is applicable to the worst scenario for an LVS (best scenario for the malicious user-vehicle).

5.3.5 Likelihood Functions and Performance Limits

In this section, the decision rule embedded in a ToA-based LVS is first formalized. In order to analyze the performance of this LVS, the likelihood functions are also derived, based on which the strategy of the malicious user-vehicle to optimally set T_x is then discussed. Finally, the performance limit of the LVS is examined.

5.3.6 Binary Decision Rule

The output of an LVS is usually a binary yes/no decision. In this chapter, the LRT is adopted as the decision rule embedded in the LVS. The LRT can achieve the minimum Total Error for the LVS. The Total Error is a combination of the false positive rate and detection rate (the *a priori* probability of a user-vehicle acting maliciously is assumed to be 0.5 in this chapter) [105]. The decision rule based on the likelihood ratio is

$$\Lambda(\mathbf{y}) \triangleq \frac{f(\mathbf{y}|\mathcal{H}_1)}{f(\mathbf{y}|\mathcal{H}_0)} \underset{D_0}{\overset{D_1}{\gtrless}} \lambda, \tag{5.11}$$

where $\Lambda(\mathbf{y})$ is the likelihood ratio, $\mathcal{D}_0$ and $\mathcal{D}_1$ are the binary decisions that infer whether the prover is genuine or malicious, respectively, and λ is the LRT decision threshold. Since the genuine and malicious user-vehicles are in similar proportions, therefore, $\lambda = 1$. A simplified Bayes average cost function is defined to quantify the performance of the LVS in terms of Total Error as below

$$\xi = p(\mathcal{H}_0)p(\mathcal{D}_1|\mathcal{H}_0) + p(\mathcal{H}_1)p(\mathcal{D}_0|\mathcal{H}_1), \tag{5.12}$$

where $p(\mathcal{H}_0)$ and $p(\mathcal{H}_1)$ are the *a priori* probabilities of occurrences of $\mathcal{H}_0$ (genuine user-vehicle), and $\mathcal{H}_1$ (malicious user-vehicle), respectively. The terms $p(\mathcal{D}_1|\mathcal{H}_0)$, and $p(\mathcal{D}_1|\mathcal{H}_1)$ are denoted as α, and β, respectively. Further, equal *a priori* probabilities for $p(\mathcal{H}_0)$ and $p(\mathcal{H}_1)$ are considered. The Eq. (5.12) takes the form

$$\xi = 0.5\alpha + 0.5(1 - \beta). \tag{5.13}$$

The decision rule given in Eq. (5.11) can achieve the minimum value of ξ, which is denoted by ξ^*.

5.4 Performance Examination on LVSs

5.4.1 Neural Network-based LVS Architecture

Finalizing an NN architecture is a crucial aspect of the NN operation. At present, this is manually done through a range of search mechanisms [87, 94]. Based on the recommendations in Chapter 2, an NN-LVS that has an input layer, a hidden layer with ten neurons, and an output layer with a binary output is finalized. This NN-LVS decides whether a user-vehicle's reported location is true or fake. The adopted NN architecture provides better accuracy and robustness. The inputs to the NN-LVS are the same as those considered for the information-theoretic LVS, *i.e.*, ToA of the transmitted signal at multiple BSs and the claimed location from a user-vehicle.

The activation function for a neuron in the hidden layer is given by

$$a_{(h,n)} = b + w_{(h-1,1)}a_{(h-1,1)} + , \ldots , + w_{(h-1,j)}a_{(h-1,j)}, \quad (n, j = 1, 2, 3, \ldots , N), \quad (5.14)$$

where b is a constant, a is the activation for a n_{th} neuron in a layer, h represents the hidden layer, j denotes the j_{th} input, and w represents the weight connecting a neuron in the input layer to the neuron in the hidden layer.

Based on numerous simulation rounds with changing transfer functions and backpropagation algorithms for the NN-LVS, the following are finalized: the hyperbolic tangent sigmoid transfer function in the hidden layer, a linear transfer function in the output layer, and the Levenberg Marquardt as the backpropagation algorithm.

The NN-LVS is supplied with training data at a speed of one random user-vehicle ToA data per second. Once the ToA data is supplied, the NN-LVS (via the backpropagation algorithm) optimizes its weights and biases through a process called 'learning'. The most frequent condition that terminates the learning process is the backpropagation algorithm's gradient descent. The gradient descent refers to minimizing the cost function through optimization of the weights and biases in

different layers of the NN-LVS as given below

$$\mathbf{w}_p \leftarrow \mathbf{w}_p - \gamma \frac{\partial}{\partial \mathbf{w}_p} J, \tag{5.15}$$

$$\mathbf{b}_p \leftarrow \mathbf{b}_p - \gamma \frac{\partial}{\partial \mathbf{b}_p} J, \tag{5.16}$$

where $\mathbf{w}_p$ is the matrix with weights for the p_{th} layer (*i.e.*, either the hidden layer or the output layer), $\mathbf{b}_p$ is a vector with baises for the p_{th} layer, γ (a dimensionless constant) is the learning rate, and J represents the calculated cost for the NN-LVS using all the weights and biases in various layers of the network. The cost 'J' is the mean square difference between the calculated output for the training data and the ground truth available with the training data.

$$J = \frac{1}{2m} \sum (\hat{\mathbf{O}} - \mathbf{O})^2,$$

where m is the number of training examples in the training data, $\hat{\mathbf{O}}$ is a vector with the calculated output for all m training samples, and $\mathbf{O}$ is a vector with the ground truth labels available for all the m training samples. The learning is terminated when $J_k \geq J_{k-1}$ over a set number of multiple iterations in a row (here this value is set to 6). The variable 'k' refers to the iteration number. Once the learning has concluded, the weights and biases for the NN-LVS are considered as tuned. The NN-LVS thereafter can be applied to the test data to calculate binary outputs in an attempt to classify the user-vehicles.

5.4.2 Effects of Bias and Assumed Threat Model

The performance for the NN-LVS is now compared with the performance of the information-theoretic framework formulated in [23]. The modelling of the channel for the malicious user-vehicle attempts to take into account the uncertainty implicit in any real-world channel for a malicious user-vehicle a distance r from a claimed location. In reality, the BS-user-vehicle channel for any user-vehicle (a malicious

or otherwise) is complex, dynamic (principally due to user-vehicle motions) and unknown. No accurate analytical theoretical model exists for such a real-world scenario. However, the key advantage of an NN-LVS over any purely information-theoretic LVS is that the former can always 'learn' the channel models for both the genuine and the malicious user-vehicles. This is true for all real-world channels irrespective of their nature (other than a purely random channel).

To make progress and to test the NN-LVS numerically, a specific real-world channel (roughly considered as a stochastic combination of bias and Rayleigh fading) is modelled for the malicious user-vehicle. It is implemented in the following manner. Each measurement by the BS is probabilistically determined as either a pure bias term or a pure scattering term (*i.e.*, no LoS component) with the weighting of each term exponentially weighted with the distance r through e^{-ar} where a is set to $\frac{1}{x}$, x being the expected distance for the malicious channel to be pure bias with probability $1/e$. Random Gaussian noise is then added to the measurement. This channel model mimics the fact that as the malicious user-vehicle approaches the claimed location, its channel model should approach that of a genuine user-vehicle, and as it moves further from the claimed location, the model provides for equal timing measurements, moduli noise. The genuine user-vehicle channel is always modelled as a pure bias channel. As will be seen in the results, the NN-LVS does indeed learn these 'real-world channels' - an outcome that persists independent of the details of the actual malicious/genuine channel model adopted.

This information-theoretic LVS calculates its binary decision by taking into account the ToA of the transmitted signal from the user-vehicle via the decision rule

$$\Lambda(\mathbf{y}) = \frac{e^{-\frac{1}{2}(\mathbf{y}-\mathbf{v})^{\mathrm{T}}\ \boldsymbol{R}^{-1}\ (\mathbf{y}-\mathbf{v})}}{e^{-\frac{1}{2}(\mathbf{y}-\mathbf{u})^{\mathrm{T}}\ \boldsymbol{R}^{-1}\ (\mathbf{y}-\mathbf{u})}} \underset{\mathcal{D}_0}{\overset{\mathcal{D}_1}{\gtrless}} \lambda, \tag{5.17}$$

where $\mathbf{v}$ ($\mathbf{v} = [v_1, v_2, v_3, \ldots, v_N]$) is the mean vector under $\mathcal{H}_1$, $\mathbf{u}$ ($\mathbf{u} = [u_1, u_2, u_3, \ldots, u_N]$) is the mean vector under $\mathcal{H}_0$, and $\mathbf{R} = \sigma_T^2 \mathbf{I}_N$ is the covariance matrix with $\mathbf{I}$ as the identity matrix.

In the comparison study shown in Fig. 5.2, four BSs in a 1000m X 500m area are

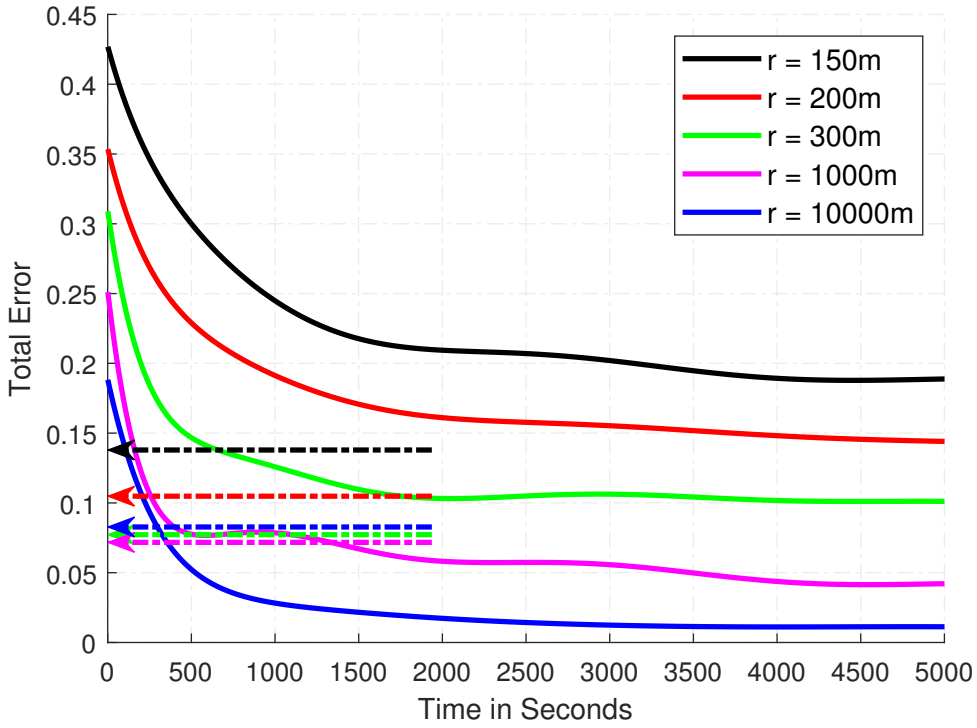

Figure 5.2: Total Error performance for NN-LVS with four BSs and changing values of r. The scenario presented here closely relates to the real-world conditions where x_i is extracted from a random Gaussian distribution, and the NLoS bias is extracted from a random exponential distribution. The standard deviations for x_i, and the NLoS bias are fixed at 100ns, and 300ns, respectively. The solid lines show the Total Error performance for the NN-LVS under different values of r, while the dashed arrow lines point to the respective Total Error calculated based on the LRT method presented in [23]. The NN-LVS reports a continuous improvement in its performance with an increasing r.

considered. The standard deviation for x_i and NLoS bias have been fixed at 100ns, and 300ns, respectively. The resultant Total Error calculated for the information-theoretic LVS based on the LRT in Eq. (5.17) is 0.14 for $r = 150$m, 0.10 for $r = 200$m, 0.07 for $r = 300$m, 0.06 for $r = 1000$m and 0.07 for $r = 10000$m. In comparison, the Total Error for the NN-LVS is 0.19, 0.14, 0.10, 0.04 and 0.01 for r equal to 150m, 200m, 300m, 1000m, and 10000m, respectively. This comparison study shows that contrary to the information-theoretic LVS of [23], the NN-LVS performs better, and its Total Error improves with increasing r. These results support the claim that the NN-LVS is efficient at learning channel conditions, has steady performance, and is more dependable in real-world situations.

The comparison study is repeated in Fig. 5.3 with a few modifications. All other

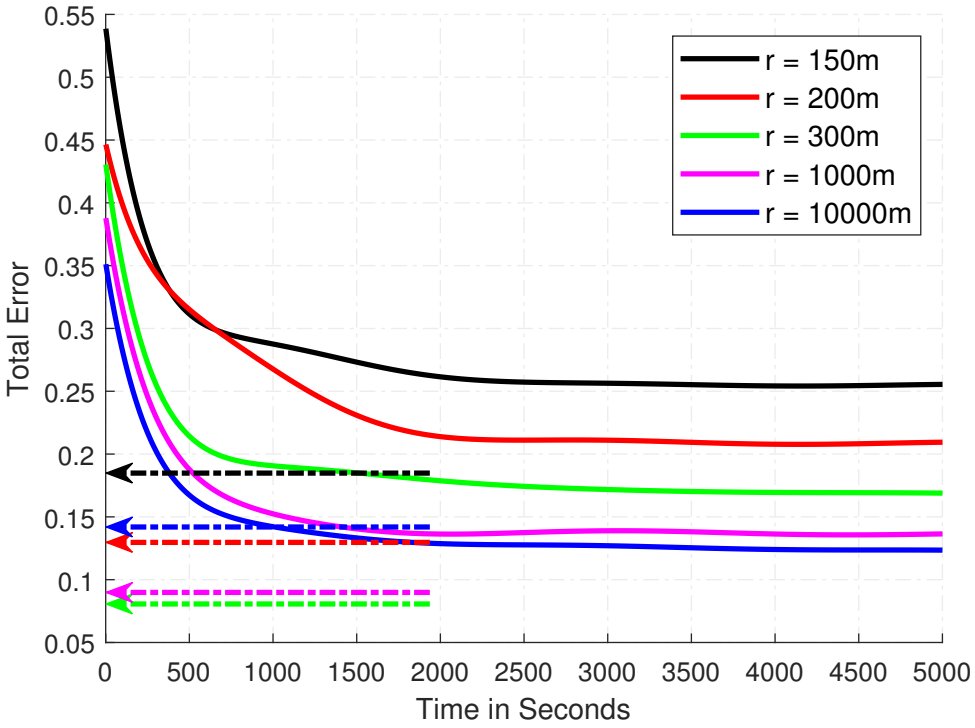

Figure 5.3: Total Error performance for the NN-LVS as in Fig. 5.2, except the standard deviations for x_i, and NLoS bias have been modified to 300ns, and 100ns, respectively.

simulation parameters are kept the same as in Fig. 5.2 except the standard deviations for x_i and NLoS bias which are modified to 300ns, and 100ns, respectively. It can be seen again that compared to the information-theoretic LVS, the NN-LVS has a better performance with increasing r.

Next, the performance of the NN-LVS is evaluated and compared with the information-theoretic LVS using a different metric, namely the Bayes Risk [121], given below

$$\mathcal{R} = p_0\, C_{00}\,(1-\alpha) + p_1\, C_{01}\,(1-\beta) + p_0\, C_{10}\,\alpha + p_1\, C_{11}\,\beta, \tag{5.18}$$

where p_0 and p_1 represent the proportion of the genuine and malicious user-vehicles. C_{00}, C_{01}, C_{10}, and C_{11} represent the different costs affecting the Bayes Risk; C_{00} is the cost associated with correctly identifying a genuine user-vehicle, C_{01} is the cost associated with falsely identifying a malicious user-vehicle as genuine (a missed detection), C_{10} is the cost associated with falsely identifying a genuine user-vehicle

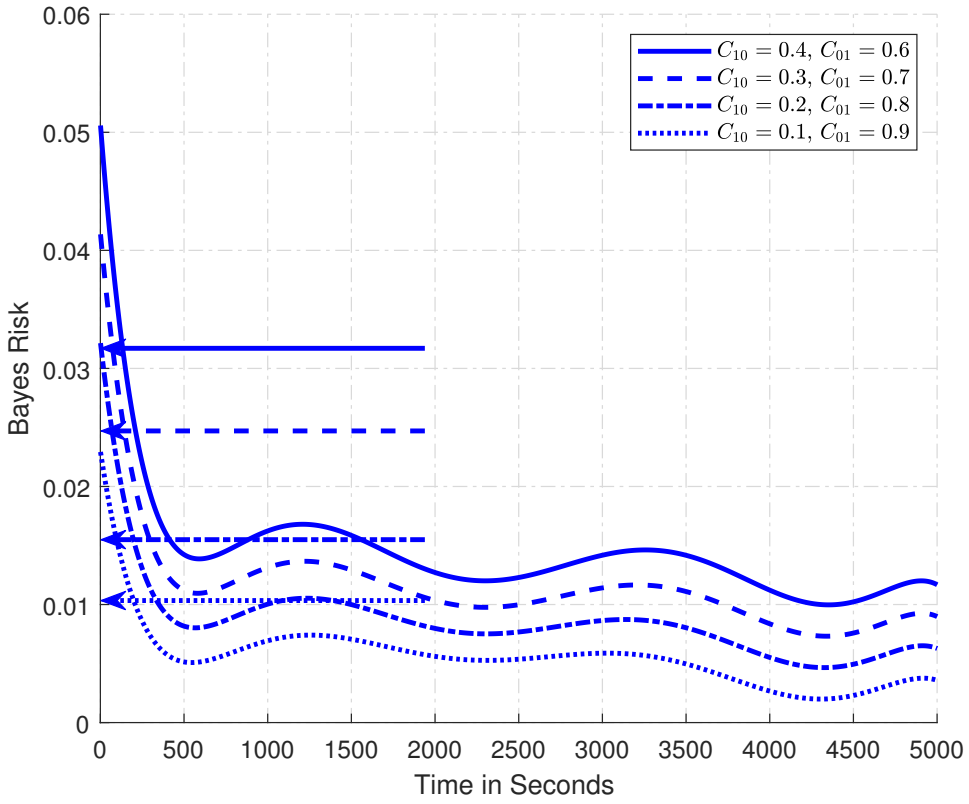

Figure 5.4: Performance evaluation for the NN-LVS and information-theoretic LVS using Bayes Risk with changing costs C_{10} and C_{01}. Here the standard deviation for x_i, and NLoS bias are 100ns, and 300ns, respectively. The value of r is set to 10,000m. The different curves represent the Bayes Risk for the NN-LVS, while the dashed arrows indicate the Bayes Risk for the information-theoretic LVS. It is evident from this figure that the NN-LVS outperforms the information-theoretic LVS in terms of Bayes Risk.

as malicious (a false positive), and C_{11} is the cost associated with correctly detecting a malicious user-vehicle. Here, C_{00} and C_{11} are considered 0 and, therefore, have no impact on the Bayes Risk. After simplification, Eq. (5.18) takes the form

$$\mathcal{R} = p_1 \, C_{01} \, (1 - \beta) \; + \; p_0 \, C_{10} \, \alpha. \tag{5.19}$$

In the real-world scenario, a missed detection, in general, is considered a more serious issue as compared to a false positive. Henceforth, a higher cost is associated with a missed detection than a false positive towards Bayes Risk calculation. With the unequal costs, the LRT decision threshold, λ, is also modified [121] and is given

by

$$\frac{f\left(\mathbf{y}|\mathcal{H}_1\right)}{f\left(\mathbf{y}|\mathcal{H}_0\right)} \underset{D_0}{\overset{D_1}{\gtrless}} \frac{p_0\left(C_{10}-C_{00}\right)}{p_1\left(C_{01}-C_{11}\right.}. \tag{5.20}$$

The variables p_0 and p_1 are assumed equal in this chapter, *i.e.*, 0.5.

The classification threshold for the NN-LVS is also modified and is set equal to λ. This means if the output of the NN-LVS for a user-vehicle (from the test set) is greater than or equal to λ, the user-vehicle is classified as malicious, else it is classified as genuine. For a wide range of changing costs for the missed detections (always more significant than the cost for false positives) the same main result was found, namely the NN-LVS outperformed the information-theoretic LVS. As an example, the performance of the two LVSs is shown by plotting the Bayes Risk with changing costs in Fig. 5.4.

5.4.3 Other Network Issues

The calculations in this chapter assume that network congestion issues do not play a role. It is assumed that the messages containing all the information required for the NN-based LVS calculations can be received (and processed) in a timely manner. If the number of user-vehicles becomes too large such that communications with the nearby BSs are interfered with, the location verification processes will be curtailed. However, eventually, it is believed an LVS will be embedded directly within the WAVE architecture as described by the IEEE 1609 suite of standards and the IEEE 8011.p standard (see [109] for discussion of all these standards). WAVE mandates the position information of all user-vehicles is broadcast every 100ms as part of the wider vehicular network safety messages, and moreover, that priority is given to these messages. The bandwidth available to such high priority messages within WAVE is designed so as to accommodate all anticipated network conditions. In this chapter, all LVS messages are assumed to be treated the same as the position information messages.

5.5 Numerical Results with the Total Error Lower Bound

The results based on Total Error Lower Bound (TLB) for both the information-theoretic LVS and the NN-LVS are now presented. The derivation of the TLB can be found in Appendix C. Four static BSs, two each at the two ends, with known true locations, are present in a 1000m X 500m area. The area in focus resembles a small district of a city and contains the claimed locations for the to-be-verified user-vehicles. For simulating the attacking scenario, two user-vehicles are considered, *i.e.*, a genuine user-vehicle reporting its true location and a malicious user-vehicle that falsifies its true location. The locations of all BSs are assumed to be known to any user-vehicle. This last assumption can be taken to mean any malicious user-vehicle can intercept all GPS information between the BSs.

As stated earlier, the malicious user-vehicle optimizes its attack location by setting T_x to T_x^* in an attempt to minimize its chances of being detected. In all the simulations, the malicious user-vehicle's forged location is at a minimum distance constraint, r, from its true location. The constraint r is an *a priori* known distance and is set to 200m in the following simulations unless otherwise specified. If the malicious user-vehicle violates the minimum distance constraint, it will be easily caught by the BSs.

Simulated ToA data of the user-vehicles transmitted signals is taken into account in this chapter. The *claimed* locations for genuine as well as malicious user-vehicles (in equal proportions) are generated randomly in the specified area, and their respective ToAs are calculated at the four BSs. The malicious user-vehicle optimizes its true location at a distance r meters away from its claimed location. The variable T_x^* is calculated by taking into account Eq. (5.10). The receivers in the BSs are under the influence of independent thermal noise x_i, and thus, the ToA measurements they make have a certain degree of variation. This variation (in nanoseconds) is extracted from a Gaussian random function that has a fixed standard deviation. As described in Eqs. (5.2) and (5.6), NLoS bias, ϕ_i, is added to the ToAs of the respec-

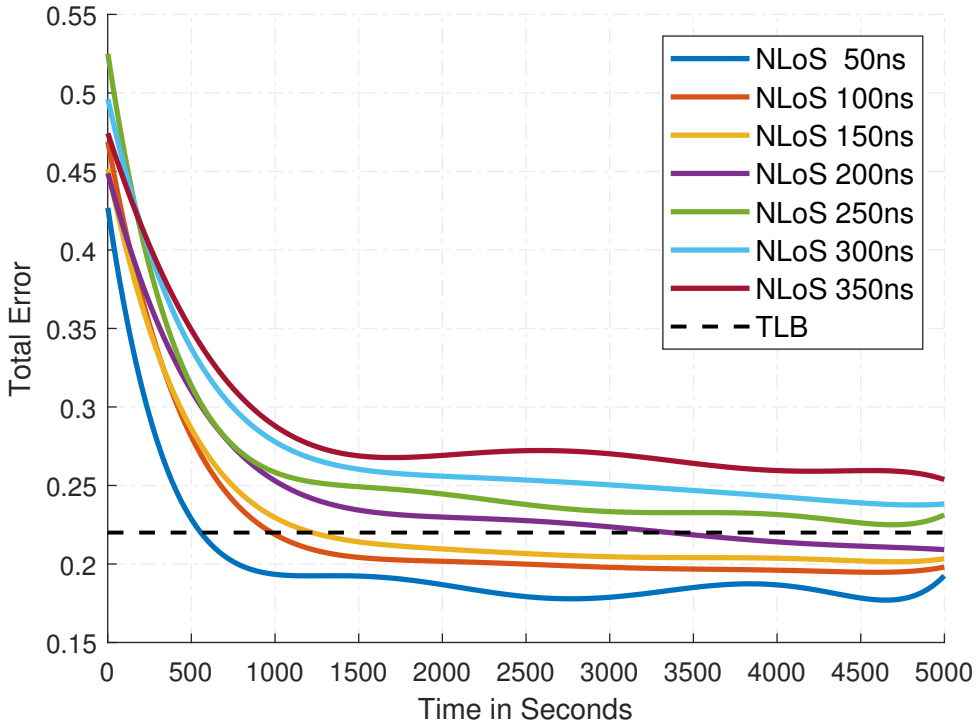

Figure 5.5: Training of the NN-LVS. The training and test datasets are simulated with four BSs. The variable x_i has a fixed standard deviation of 300ns. The value of r is fixed at 200m. Standard deviation for NLoS bias is changing in training and test datasets simultaneously and is represented by different colour of curves in the figure. TLB is the information-theoretic TLB. The Total Error for the NN-LVS increases with an increase in the NLoS bias.

tive user-vehicles. To mimic reality, the NLoS bias is extracted from an exponential distribution with a fixed standard deviation, as highlighted in Eq. (5.4).

The TLB is now taken into account in the study in Fig. 5.5. The TLB is calculated using equation (C.10). In the TLB calculation, r is equal to 200m while the standard deviation for x_i, and NLoS bias is 300ns, and 900ns, respectively. This value of standard deviation for NLoS bias is assumed to be closely related to the global average for NLoS[1]. The calculated TLB with four BSs and the above settings is 0.22.

[1]It is assumed that that *a priori* information on the channel conditions over some global average is available here. The TLB then is used to provide a new stopping condition under a range of differing real-world test data. As will be seen, there will be some test data where the TLB has no effect (usual stopping condition applies), and other test data where the stopping condition is reached well before the usual stopping condition. Even though the new training time is only about $\frac{1}{5}$ of the usual training time, the performance 'hit' is minimal.

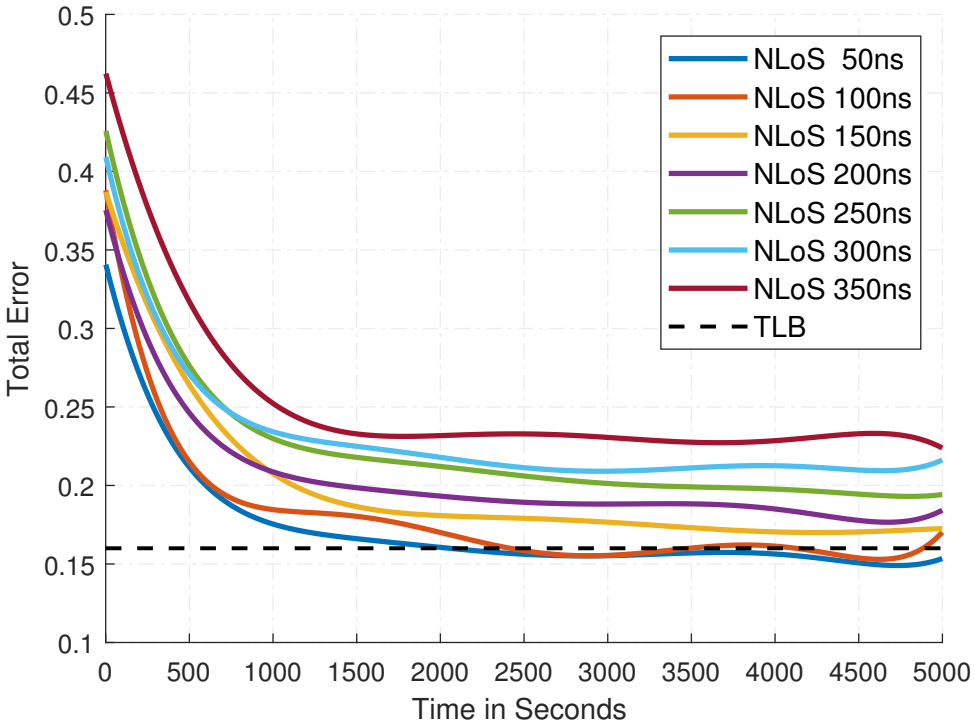

Figure 5.6: Training of the NN-LVS as in Fig. 5.5 except the number of BSs in this simulation has been modified to six.

For training the NN-LVS in Fig. 5.5, a training dataset simulated with an NLoS bias standard deviation of 50ns is utilized. Other simulation parameters are the same as those considered for the TLB calculations earlier. The training data has genuine and malicious user-vehicles in equal proportions. Before training the NN-LVS, the training data is randomized to closely mimic the reality. The NN-LVS is trained with the random user-vehicle data in each second and is further used to calculate a Total Error for the test data. Before re-training the NN-LVS and calculating a revised Total Error for the test data in each subsequent second, the previous training data is accumulated with a new random user-vehicle training data from that second. Total Error for the test data is plotted against time as shown in Fig. 5.5 (a polynomial fitting is applied). The figure also has Total Error curves for different values of NLoS bias standard deviation.

In Fig. 5.6, the same simulation as in Fig. 5.5 is carried out, but the number of BSs is revised to six. None of the other simulation parameters is changed. One can see that the NN-LVS with six BSs performs better than the NN-LVS with four

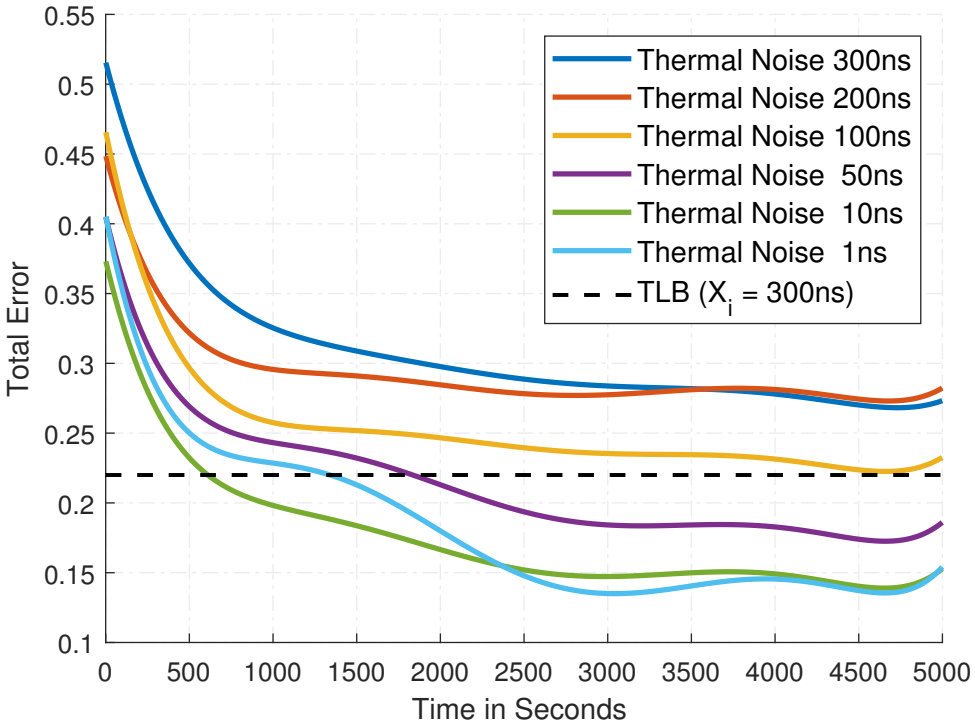

Figure 5.7: NN-LVS performance with changing values of receivers thermal noise in the BSs. Other parameter settings in this simulation are: Standard deviation for NLoS is 400ns and the number of BSs is four. It can be seen that NN-LVS performs better with low receiver thermal noise.

BSs. This highlights the fact that more BSs will result in better performance for the NN-LVS.

In Fig. 5.7, the impact of changing the BS's thermal noise, x_i, over the Total Error for the NN-LVS is studied. It is evident from this figure that NN-LVS shows improved performance at lower values of x_i. The TLB in Fig. 5.7 has been calculated with x_i = 300ns, NLoS bias equal to 900ns and r = 200m. We see the case of x_i = 300ns represents a scenario where the TLB would not be used as the stopping condition.

Several numerical simulations of the trained NN-LVSs have been carried out as applied to a wide range of data simulating real-world deployment. The performance in term of the Total Error of the NN-LVSs in these tests shows similar results to the flat parts of the curves shown in Figures 5.5–5.7. That is, the designed NN-LVSs, once trained, perform as expected in our mimicked real-world deployments. In a nutshell, we see that when the NN-LVS training is stopped based on the TLB, we,

on average, can cut the training time by a factor of $\frac{1}{5}$. That is, we can save training data costs and ensure quick deployment of the NN-LVS with minimal impact on the overall performance.

5.6 Conclusion

A new system for solving the critical problem of location verification in the context of vehicular networks has been presented. Encompassing information theory and NN concepts, the solution that has been provided will lead to enhanced and pragmatic location verification outcomes. Of particular importance is the fact that new scenarios have been covered to show how an NN-based LVS is able to outperform a pure information-theoretic LVS when channel conditions are *a priori* unknown and/or the malicious user-vehicle is at an unspecified distance from its claimed location. Furthermore, it has also been shown how optimal-information-theoretic concepts can be encapsulated with the NN framework so as to allow for a very useful trade-off in the training-time *vs.* performance - thus making the designed solution even more practical in real-world deployments.

The work in this chapter can also facilitate a quick deployment location verification solution for many other location-oriented applications within IoT (such as intelligent parking system, map services, smart supply chain and logistics, etc.) and for enhancing the efficiency of numerous features within future wireless telecommunication systems (e.g., beamforming, mMIMO for enhanced throughput, and interference mitigation for quality and capacity improvements).

NNs designed for location verification can perform to specific limits under given channel circumstances. In the next chapter, an NN framework is formulated by combing multiple standalone NNs. The new NN framework can improve its performance beyond the performance of the individual NNs.

Chapter 6

Combining Different Neural Networks for Location Verification

In the previous chapter, a very important performance parameter of the NN-LVS, *i.e.*, the training time was investigated. Even with optimum parameters setting, such frameworks at best can perform to certain limits. This chapter looks into the design of new NNFs for location verification. These NNFs are able to improve the location verification performance beyond those of the individual NN-LVSs.

Several NN-based LVS solutions [28, 97, 98] were formulated in the previous chapters to address operational limitations of the traditional LVSs [45, 47, 89–91]. These NN-based LVS solutions can adequately perform in a changing environment if re-trained with new training data (that represents the varied channel circumstances). Moreover, even if re-trained, such solutions have limited performance. All these constraints disfavour the deployment of such a solutions real-world settings. This chapter utilizes a pooling concept in combining two standalone Feedforward Neural Networks (FNNs) to design intelligent NNFs for location verification. The consideration of combining multiple FNNs for formulating the NNFs relate to the number of attacking strategies that a malicious vehicle can take into account while launching a location spoofing attack. For example, the malicious vehicle at any given time can adopt one of the two location spoofing strategies, *i.e.*, random or

optimized.

Approaches similar to the ones designed here have been adopted in a few other NN application areas, *i.e.,* water quality monitoring [122], computer vision [123], and language identification [124], etc. What makes the NNFs design in this chapter different from the approaches adopted in other NN application areas is that the NNFs do not utilize both the FNNs simultaneously. In one scenario, an NNF formulates its parameters (to be used for vehicle classification) by taking the weighted average of the parameters of the trained FNNs. In another scenario, a second NNF, at any given time, makes use of anyone (and not both) of the trained FNNs for vehicle classification. The individual FNNs are trained with data realized under different channel conditions. One of the NNFs utilizes the existing training of the FNNs and consider a statistical approach to enhance the location verification of vehicles' claimed locations in VANETs. For comparison purposes, a selective strategy is also taken into account in formulating a second NNF that is different from the previously designed NNF. Both the NNFs consider the RSS measurements of the vehicles' transmitted signals for their operation. The NNFs show a consistent performance under changing threat situations and report a significant performance boost compared to the individual FNNs in some cases. It is believed that the new approach will prove vital in location-centric applications within other wireless networks where location verification is of paramount importance.

6.1 Introduction

The performance of the current and future wireless networks will highly depend on the location information of the users. Such information is usually user-dependent - meaning that the users acquire their location information with the aid of a satellite-based system and provide it to the wireless network once requested. A possibility exists where a user attempts to spoof the system and provides its wrong location information to the network. Alternately, the user may supply the wrong location information due to some software or hardware issues. The consequences of this

can be dire for the wireless network if errors in such information go unnoticed. Verification of the location information is hence crucial before using it for the network operation.

The authentication of the users' reported location information has attracted considerable research attention in recent years. We find some state-of-the-art heuristic [27, 58–60, 125–127] as well as information-theoretic [24–26, 79, 81] frameworks to address the problem of location verification in wireless networks. The problem around such frameworks is that they take into account *a-priori* knowledge on key channel parameters at the time of their design. However, the assumed channel conditions most likely do not relate to the actual channel conditions at the time and place of deployment. An example is the design of an LVS aimed to function in an urban area. While this LVS may work well in the designated area, it may not function in a suburban area. Moreover, the same LVS may not perform as desired in the specified area in harsh weather situations. To overcome such limitations in heuristic as well as information-theoretic LVSs, NN algorithms have been considered in formulating intelligent location verification solutions [28, 97, 98].

Although flexible and practical, the NN-LVS solutions can perform to specific thresholds under given channel settings. Moreover, an NN-LVS trained to address a particular attack (threat model) will need re-training to manage new threat scenarios. One example, and that which will form the focus of this chapter, is an NN-LVS trained to detect malicious vehicles claiming their locations randomly at a certain distance away from their actual locations. The NN-LVS can perform to a limit and with some error in classifying the vehicles in the area. Suppose the vehicles in the area now acquire new resources and start optimizing their claimed locations to minimize their chances of being detected. In such a scenario, the performance of the NN-LVS will likely be poor. To accommodate the changed threat conditions and classify the vehicles within acceptable performance thresholds, we have two options; either deploy a new NN-LVS trained to handle the changed attack situation or re-train the existing NN-LVS to cope with the changed threat conditions. Such periodical re-training may not always be possible in real-world situations. Therefore, there is a need for an intelligent LVS framework which not only closes the opera-

tional limitations of the traditional LVS algorithms, but can remove the re-training requirements of the earlier designed NN-LVSs – whilst performing efficiently.

The main purpose of this chapter is to design novel NNFs for NN-LVSs that can accommodate different threat models, and varying channel conditions, by dynamically altering the underlying structure of the framework based on some new input 'signal' that is gathered in the field. It is believed that the new approach described here in accommodating unknown threat models for ML-based security functions to be entirely novel even in the wider sphere of generic network intrusion detection. As the RSS of a vehicle's transmitted signals, measured at multiple static BSs will form the focus of the specific LVSs to be investigated here, this new input signal will be related to those RSS values. To add additional focus, the study in this chapter will be carried out in the context of VANETs, however, the concepts discussed are widely applicable to many other wireless networks. A summary of the main contributions in this chapter are as follows:

1. Two separate FNNs are designed and merged to formulate new NNFs for location verification in VANETs. These NNFs, unlike the earlier formulated LVSs, are able to manipulate the unusual channel conditions and unknown threat models.

2. Through simulated data it is shown that not only are these novel NNFs consistent in their performance but can significantly improve the overall location verification performance relative to individual FNNs.

The remainder of this chapter is organized as follows. Section 6.2 details the system model. Section 6.3 presents the NNF. Section 6.4 provides numerical results, and Section 6.5 concludes this chapter.

6.2 System Model

The following system model is considered in this chapter:

1. The true location of a random vehicle, $\mathbf{x}_t = [\mathrm{x}_t, \mathrm{y}_t]$, is unknown to the framework.

2. The reported or announced location from a legitimate or malicious vehicle, $\mathbf{x}_c = [\mathrm{x}_c, \mathrm{y}_c]$, is termed as 'claimed location'.

3. The area under consideration has N BSs. The true locations of the BSs, which have zero localization error in them, is publicly known to the random vehicles in the area. The true location of the i-th BS is denoted by $\mathbf{x}_i = [\mathrm{x}_i, \mathrm{y}_i]$, where $i = 1, 2, 3, \ldots, N$.

4. The BSs in the area can communicate with the surrounding vehicles. Moreover, each BS independently measures the RSS (all RSS in dBm) of the vehicles transmitted signals at a frequency of one RSS measurement per vehicle per second.

5. One of the BSs is under the influence of thermal noise and impacts the RSS measurements here. This thermal noise is extracted from a random Gaussian distribution that has a fixed standard deviation σ_t.

6. One of the N BSs is chosen as the PC. The PC performs its usual task of measuring the RSS from the random vehicles in the area and collects the RSS measurements from the surrounding BSs. The PC combines its own RSS measurements with those collected and processes them to issue instructions for the network operation.

7. Under the null hypothesis $\mathcal{H}_0$, the framework assumes a vehicle to be legitimate, *i.e.*, $\mathcal{H}_0 : \mathbf{x}_c = \mathbf{x}_t$. A log-normal shadowing model is adopted for the RSS observations. Under $\mathcal{H}_0$, the measured RSS at the i-th BS, *i.e.*, y_i, is given as

$$y_i = u_i + X_{\sigma_{db_i}}, \qquad (i = 1, 2, 3, \ldots, N), \tag{6.1}$$

where $X_{\sigma_{db}}$ is a zero-mean normal random variable with variance σ^2_{db} representing the shadowing noise, and u_i is given as

$$u_i = P_t - \Big[P_{d_o} + 10\,\gamma\,\log_{10}\Big(\frac{d_i^c}{d_0}\Big)\Big], \tag{6.2}$$

where P_t is the transmit power of the random vehicle, P_{d_0} is the measured RSS at a reference distance d_0, γ is the path loss exponent, and d_i^c ($d_i^c > d_0$) is the distance between the i_{th} BS and the legitimate vehicle's claimed location (which is also his true location). This distance is given by $d_i^c = \sqrt{(\mathrm{x}_i - \mathrm{x}_c)^2 + (\mathrm{y}_i - \mathrm{y}_c)^2}$. The RSS measurements made by the N BSs are independent of each other. They collectively form an RSS vector given by $\mathbf{y} = [y_1,\ y_2,\ y_3, \ldots, y_N]$.

8. Under the alternate hypothesis $\mathcal{H}_1$, the framework considers a vehicle to be malicious, *i.e.*, $\mathcal{H}_1 : \mathbf{x}_c \neq \mathbf{x}_t$. Under $\mathcal{H}_1$, the RSS measurement at the i-th BS, *i.e.*, y_i, is given as

$$y_i = v_i + X_{\sigma_{db_i}}, \tag{6.3}$$

where v_i is given as

$$v_i = P_t - \Big[P_{d_o} + 10\,\gamma\,\log_{10}\Big(\frac{d_i^t}{d_0}\Big)\Big], \tag{6.4}$$

where d_i^t is the distance of the i_{th} BS to the true location of the malicious vehicle and is given by $d_i^t = \sqrt{(\mathrm{x}_i - \mathrm{x}_t)^2 + (\mathrm{y}_i - \mathrm{y}_t)^2}$. The RSS measurements made by the N BSs are independent of each other. They collectively form an RSS vector given by $\mathbf{y} = [y_1,\ y_2,\ y_3, \ldots, y_N]$.

To spoof the network, a malicious vehicle can launch an attack by claiming its location randomly, at least, r meters away from its true location. Alternately, the malicious vehicle has all the resources to optimize its claimed location in an attempt to minimize its chances of being detected. To simulate the latter scenario, the malicious vehicle minimizes its KL divergence from $f(\mathbf{y}|\mathcal{H}_1)$ to $f(\mathbf{y}|\mathcal{H}_o)$ [108] as

below [24, 107]

$$D_{KL}(f(\mathbf{y}|\mathcal{H}_1)||f(\mathbf{y}|\mathcal{H}_0)) = \int_{-\infty}^{\infty} f(\mathbf{y}|\mathcal{H}_1) \ln \frac{f(\mathbf{y}|\mathcal{H}_1)}{f(\mathbf{y}|\mathcal{H}_0)} d\mathbf{y}, \tag{6.5}$$

$$= \frac{1}{2}(\mathbf{v}-\mathbf{u})^T \boldsymbol{\Sigma}^{-1}(\mathbf{v}-\mathbf{u}), \tag{6.6}$$

where $\mathbf{v} = [v_1,\ v_2,\ v_3, \ldots, v_N]^\top$, $\mathbf{u} = [u_1,\ u_2,\ u_3, \ldots, u_N]^\top$, and $\boldsymbol{\Sigma} = \sigma_{db}^2 \mathbf{I}_N$ is the covariance matrix with $\mathbf{I}$ as the identity matrix. The size of $\mathbf{I}$ is $N \times 1$. Following Eqs. (6.5) and (6.6) , the optimal claimed location $\mathbf{x}_c^*$ for the malicious vehicle can be obtained through

$$\mathbf{x}_c^* = \underset{|\mathbf{x}_t - \mathbf{x}_c| \geq r}{argmin}\, D_{KL}(f(\mathbf{y}|\mathcal{H}_1)||f(\mathbf{y}|\mathcal{H}_0)). \tag{6.7}$$

6.3 Neural Network Frameworks

This section highlights the architecture and methodologies taken into account for formulating the NNFs. Each of the NNFs considered in this work is a combination of two independent but identical FNNs.

6.3.1 The Architecture of the Feedforward Neural Network

The internal architecture of one of the two identical FNNs utilized to form the NNFs is now explained. An FNN is a particular type of an NN that is known to manipulate and learn from the physical layer properties of the vehicles transmitted signals [28, 97, 98]. Keeping in view the number of features in the input data, we start with a single hidden layer for the FNN and find it sufficient for the type of channel under investigation in this work [128]. We find that adding more hidden layers marginally improves the FNN's performance at the cost of an exponential rise in the computational requirements, and the training time. Moreover, adding more hidden layers unnecessarily may also take the FNN close to over-fitting. As such, we do not recommend more than one hidden layer for the FNN. The architecture of the FNN may vary once the channel between the vehicles and BSs has additional

noises and biases in it.

The output value of the FNN is either 1 (malicious) or 0 (legitimate). Therefore, a tangent sigmoid transfer function, *i.e.*, $a(x) = (1 - e^{-2x})(1 + e^{-2x})^{-1}$, is used in the hidden layer and a logistic sigmoid transfer function, *i.e.*, $a(x) = (1 + e^{-x})^{-1}$, is utilized in the output layer of the FNN. This choice of the transfer functions is advised for the FNNs used for binary classification, in general. The number of neurons in the hidden layer is kept in line with the range advised in the Chapter 2, *i.e.*, ten [128].

The two FNNs (to form the NNFs) are named NN_R and NN_O. Separate training sets simulated under different channel conditions are used to train the NN_R and the NN_O. Each training set comprises of the RSS measurements, the claimed locations for all K vehicles in the training set, the expected RSS (derived using the vehicles' claimed locations), a feature $h_{1,k} = \sum_{i=1}^{N} |y_{meas_{k,i}} - y_{exp_{k,i}}|$ (where $y_{meas_{k,i}}$, and $y_{meas_{k,i}}$ are the measured, and expected RSS for the k_{th} vehicle on the i_{th} BS in the training set, respectively), and a feature $h_{2,k} = \frac{1}{N} h_{1,k}$. The training sets considered are assumed to have genuine and malicious vehicles in equal proportions unless otherwise specified (see [28] for a comprehensive study detailing the performance of the FNN with modified ratios of genuine and malicious vehicles). The NN_R is trained with a training set where all the malicious vehicles randomly claim their locations. Similarly, the NN_O is trained with a different training set where all the malicious vehicles optimally claim their locations. The trained NN_R and NN_O are next combined to form the joint NNF. A schematic of the NNF is shown in Fig. 6.1. This architecture of the NNF is preferred over a standalone FNN trained with data that has a mixture of malicious vehicles with random and optimized attacking strategies (the trained standalone FNN performs poorly when subject to testing).

6.3.2 Weighted Neural Network Framework

This section highlights the working methodology of an NNF named the Weighted Neural Network Framework (WNNF). To classify the m_{th} test sample (where $m =$

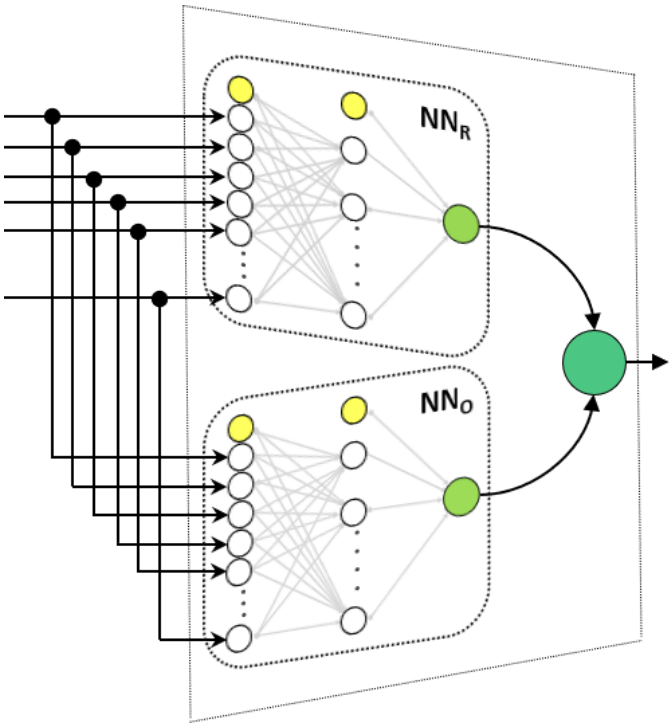

Figure 6.1: The architecture of the NNF. The NNF is formed through the combination of two identical FNNs namely NN_R and NN_O.

1,2,3,..., M, with M being the total number of samples in the test set), the WNNF statistically weights the two FNNs (*i.e.*, NN_R, and NN_O) based on a variable z_m (a real number). The variable z_m assigns weights to the optimized parameters of the FNNs used to form the WNNF. The variable z_m is defined as

$$z_m = \frac{\zeta}{y_m}, \tag{6.8}$$

where $\zeta = \frac{1}{N_v N} \sum_{n=1}^{N_v} \sum_{i=1}^{N} |y_{meas_{n,i}} - y_{exp_{n,i}}|$ and $y_m = \frac{1}{N} \sum_{i=1}^{N} |y_{meas_{m,i}} - y_{exp_{m,i}}|$. Here, N_v is the number of malicious vehicles in the training set, while $y_{meas_{n,i}}$, and $y_{exp_{n,i}}$ is the measured, and expected RSS for the n_{th} malicious vehicles at the i_{th} BS in the training set used to train the NN_O, respectively. The variables $y_{meas_{m,i}}$, and $y_{exp_{m,i}}$ are the measured, and expected RSS for the m_{th} test set sample at the i_{th} BS, respectively. The RSS $y_{exp_{m,i}}$ for the m_{th} test sample is calculated using the claimed location of a vehicle at the i_{th} BS. From the definition, we see that ζ will be the average RSS difference that we will observe when a malicious vehicle claims to be approximately r meters away from its true location. It is known that a malicious vehicle, while launching an attack, always claims its location at a distance of r

meters or greater (depending on whether the malicious vehicle claims its location randomly or through optimization) from its true location. This means that $\mathrm{y}_m \geq \zeta$ for a vehicle that is malicious. Hence, the value of z_m will be in the range 0 – 1 for such a vehicle. On the other hand, genuine vehicles will mostly have $\mathrm{y}_m < \zeta$, which means that $z_m > 1$. The normalized value of the variable z_m is used to calculate the weights and biases in each layer of the WNNF as below:

$$\mathbf{b}_{\eta_m}^{[1]} = \mathrm{z}_m * \mathbf{b}_{\Upsilon}^{[1]} + (1 - \mathrm{z}_m) * \mathbf{b}_{\psi}^{[1]}, \tag{6.9}$$

$$\mathbf{W}_{\eta_m}^{[1]} = \mathrm{z}_m * \mathbf{W}_{\Upsilon}^{[1]} + (1 - \mathrm{z}_m) * \mathbf{W}_{\psi}^{[1]}, \tag{6.10}$$

$$\mathbf{b}_{\eta_m}^{[2]} = \mathrm{z}_m * \mathbf{b}_{\Upsilon}^{[2]} + (1 - \mathrm{z}_m) * \mathbf{b}_{\psi}^{[2]}, \tag{6.11}$$

$$\mathbf{W}_{\eta_m}^{[2]} = \mathrm{z}_m * \mathbf{W}_{\Upsilon}^{[2]} + (1 - \mathrm{z}_m) * \mathbf{W}_{\psi}^{[2]}, \tag{6.12}$$

where the superscript '$^{[1]}$' refers to the connections between the input layer and the hidden layer of an FNN. Similarly, the superscript '$^{[2]}$' refers to the connections that connect the FNN's hidden layer to the FNN's output layer. The notations η, ψ, and Υ, represents the WNNF, the NN_R, and the NN_O, respectively. Following the above discussion, the parameters in $\mathbf{b}_{\eta_m}^{[1]}$, and $\mathbf{W}_{\eta_m}^{[1]}$ represent the biases and weights of the WNNF between its input and hidden layers used to classify the m_{th} test sample, respectively. Similarly, $\mathbf{b}_{\eta_m}^{[2]}$, and $\mathbf{W}_{\eta_m}^{[2]}$ are the WNNF's biases, and weights connecting its hidden and output layers utilized to classify the m_{th} test sample, respectively. $\mathbf{b}_{\Upsilon}^{[1]}$, and $\mathbf{W}_{\Upsilon}^{[1]}$, are the biases, and weights that connect the input layer of the NN_O to the hidden layer of the NN_O, respectively. Similarly, $\mathbf{b}_{\Upsilon}^{[2]}$, and $\mathbf{W}_{\Upsilon}^{[2]}$, are the biases, and weights connecting the hidden layer of the NN_O to the output layer of the NN_O, respectively. For the NN_R, $\mathbf{b}_{\Psi}^{[1]}$, and $\mathbf{W}_{\Psi}^{[1]}$, are the biases, and weights used to connect its input and hidden layers. Similarly, $\mathbf{b}_{\Psi}^{[2]}$, and $\mathbf{W}_{\Psi}^{[2]}$, are the biases, and weights connecting its hidden and output layers, respectively. Both $\mathbf{W}_{\Upsilon}^{[1]}$ and $\mathbf{W}_{\Psi}^{[1]}$ have dimensions $n_h \times n_x$. The bias vectors $\mathbf{b}_{\Upsilon}^{[1]}$ and $\mathbf{b}_{\Psi}^{[1]}$ each has dimensions $n_h \times 1$. The weights $\mathbf{W}_{\Upsilon}^{[2]}$ and $\mathbf{W}_{\Psi}^{[2]}$ both are of dimensions $n_y \times n_h$. The dimensions of $\mathbf{b}_{\Upsilon}^{[2]}$ and $\mathbf{b}_{\Psi}^{[2]}$ are the same, *i.e.*, $n_y \times 1$. In this chapter, $n_x = 14$, $n_h = 10$, and $n_y = 1$. Finally, the output of the WNNF for the m_{th} test sample is derived as, $O_{\eta_m} = \boldsymbol{\eta}_m(\mathbf{X}_m)$, where $\boldsymbol{\eta}_m$ is a dictionary containing all the weights and biases

in various layers of the WNNF which are used to classify the m_{th} test sample. The vector $\mathbf{X}_m$ represents the input features for the m_{th} test sample and has dimensions $n_x \times 1$, *i.e.*, $\mathbf{X}_m = [X_{m,1},\ X_{m,2},\ X_{m,3}, \ldots, X_{m,n_x}]$.

6.3.3 Selective Neural Network Framework

It would be useful to investigate the location verification of the vehicles through a new strategy that is different from the WNNF. This new strategy combines the NN_R and the NN_O into a joint framework that we call the Selective Neural Network Framework (SNNF). Based on $\mathbf{X}_m$, the SNNF, at any given time, will choose one of the two FNNs (*i.e.*, either the NN_R, or the NN_O) for vehicle classification. The SNNF utilizes the value z_m in Eq. (6.8) to define an operational boundary between the NN_R and the NN_O. The variable z_m is expected to be ≥ 1 for malicious vehicles. Similarly, for genuine vehicles, $z_m < 1$. As the training sets used to train the NN_R and the NN_O both have data for genuine vehicles in them, any of these two FNNs can be utilized to classify such vehicles in the test set. This chapter considers the NN_O for classifying vehicles where $z_m > 1$. Moving forward, the variable z_m is rounded to the nearest integer. All the values of $z_m \geq 1$ are set to 1, *i.e.*, $z_m(z_m \geq 1) = 1$. Finally, the following equation is taken into account for the SNNF to classify the m_{th} test sample

$$\mathrm{z}_m \underset{NN_R}{\overset{NN_O}{\gtrless}} 1. \tag{6.13}$$

6.3.4 Performance Criterion

A Bayes average cost function is taken into account as the performance metric for the FNNs and NNFs. This performance metric is named the 'Total Error' (ξ). Mathematically

$$\xi = p(\mathcal{H}_0)\alpha + p(\mathcal{H}_1)(1 - \beta), \tag{6.14}$$

where $p(\mathcal{H}_0)$, and $p(\mathcal{H}_1)$ relates to the *a priori* probabilities of occurrences of $\mathcal{H}_0$, and $\mathcal{H}_1$, respectively, α is the false positive rate, and β represents the detection rate. The false positive rate (*i.e.*, α) is defined as the probability of rejecting $\mathcal{H}_0$, while the detection rate (*i.e.*, β) is the probability of accepting $\mathcal{H}_1$. Since, the training set has genuine and malicious vehicles in equal proprotion, both $p(\mathcal{H}_0)$ and $p(\mathcal{H}_1)$ are set equal to $\frac{1}{2}$. Equation (6.14) thus simplifies to

$$\xi = \frac{1}{2}\alpha + \frac{1}{2}(1-\beta). \tag{6.15}$$

6.3.5 Avoiding Over-training

An over-training of an FNN is when the FNN is trained to such a limit that it memorizes the data in the training set. An over-trained FNN will perform beyond optimal limits if exposed to the same data as in the training set but will perform poorly on any unseen data. An information-theoretic framework is known to achieve the best ξ under idealistic channel situations [105]. To mimic such idealistic channel conditions, the added bias due to the thermal noise of the BS is set to zero. The value for ξ is determined using information-theoretic expressions and compared with ξ_{NN_R} and ξ_{NN_O} to check if the NN_R and NN_O are over-trained. The information-theoretic framework considers the LRT to calculate ξ [79] as below

$$\Lambda(\mathbf{y}) \triangleq \frac{p(\mathbf{y}|\mathcal{H}_1)}{p(\mathbf{y}|\mathcal{H}_o)} \underset{\mathcal{D}_0}{\overset{\mathcal{D}_1}{\gtrless}} \lambda, \tag{6.16}$$

where $\Lambda(\mathbf{y})$ represents the likelihood ratio, $p(\mathbf{y}|\mathcal{H}_1)$ and $p(\mathbf{y}|\mathcal{H}_0)$ are the probability density functions of the RSS vector under $\mathcal{H}_1$ and $\mathcal{H}_0$, λ is the decision threshold and is calculated as the ratio of the probabilities of occurrences of $\mathcal{H}_1$ (*i.e.*, malicious vehicle) and $\mathcal{H}_o$ (*i.e.*, legitimate vehicle), and $\mathcal{D}_1$ and $\mathcal{D}_0$ relate to the binary decision values (*i.e.*, whether the vehicle to be classified is malicious or legitimate). Considering the multivariate normal form of the observations and the fact that we have genuine and malicious vehicles in equal proportions, Eq. (6.16) can be reformulated

as [25]

$$\Lambda(\mathbf{y}) = \frac{e^{-\frac{1}{2}(\mathbf{y}-\mathbf{v})^{\mathrm{T}}\ \Sigma^{-1}\ (\mathbf{y}-\mathbf{v})}}{e^{-\frac{1}{2}(\mathbf{y}-\mathbf{u})^{\mathrm{T}}\ \Sigma^{-1}\ (\mathbf{y}-\mathbf{u})}} \underset{\mathcal{D}_0}{\overset{\mathcal{D}_1}{\gtrless}} 1. \tag{6.17}$$

The expression to calculate ξ for the information-theoretic framework is given as

$$\xi = p(\mathcal{H}_0)\, p(\mathcal{D}_1|\mathcal{H}_0) + p(\mathcal{H}_1)\, p(\mathcal{D}_0|\mathcal{H}_1), \tag{6.18}$$

where $p(\mathcal{D}_1|\mathcal{H}_0)$, and $p(\mathcal{D}_0|\mathcal{H}_1)$ are α, and $(1-\beta)$, respectively.

To validate that the NN_R and the NN_O are not over-trained, a channel environment where $N = 5$, $\gamma = 3$, $\sigma_{db} = 3$ dB, $\sigma_t = 0$ dB, and $r = 50$ meters is considered. The number of epochs for training the NN_R and NN_O is set to 4000. With the given channel parameters, ξ is calculated for a scenario where the malicious vehicles randomly claim their locations. The derived values for ξ for the information-theoretic framework, and the NN_R are equal to 0.03, and 0.07, respectively. Considering the same parameter settings as above, a different channel scenario is assumed where the malicious vehicles always optimize their claimed locations. The revised ξ for the information-theoretic framework, and the NN_O are 0.15, and 0.22, respectively. This proves that the two FNNs (*i.e.*, the NN_R and the NN_O) are not over-trained.

6.4 Numerical Results

This section presents the numerical results. For focus, a 200×200 square meters area – mimicking a highway junction is considered. The area has 5 BSs installed at (0m, 0m), (0m, 200m), (100m, 100m), (200m, 0m), and (200m, 200m). These BSs independently measure RSS from random vehicles in the focus area. The shadowing noise elements for the individual BSs in Eq. (6.1) are extracted from a random Gaussian distribution with a fixed standard deviation σ_{db}. The bias element (highlighted in Section 6.2 (5) as the internal thermal noise) at one of the BS is derived from a different random Gaussian distribution that has a fixed standard deviation, *i.e.*, σ_t, equal to 2 dB. The parameter values for path loss exponent γ, and minimum

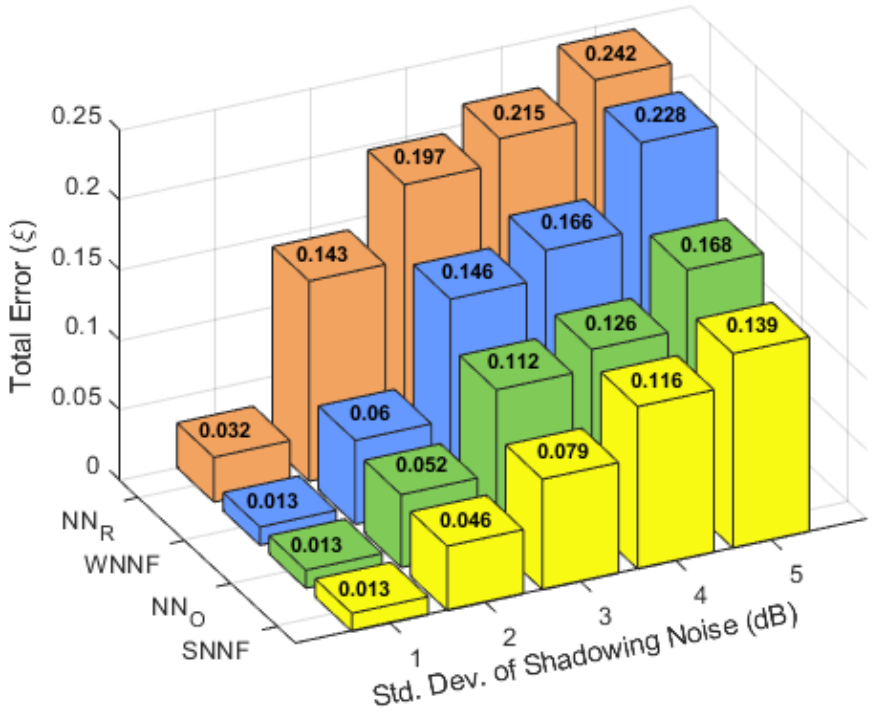

Figure 6.2: A comparison of the performances of the WNNF and the SNNF with the two FNNs (*i.e.*, NN_R, and NN_O) under a channel situation where σ_{db} varies. All the other channel parameter settings are fixed. Overall, we see that the SNNF improves ξ, in comparison to ξ, of the individual FNNs, and the WNNF.

distance r are set to 3, and 50 meters, respectively in Eq. (6.2). At the end of the RSS measurement campaign, the database has 10,000 samples. These samples are divided into equal proportions of genuine and malicious vehicles. The samples representing genuine vehicles have associated true location coordinates, while those representing malicious vehicles have associated spoofed location coordinates. Each sample representing inputs (*i.e.*, RSS measurements and location coordinates) for a genuine vehicle is assigned a label '0'. Similarly, each sample representing inputs for a malicious vehicle is assigned a label '1'. The database is randomized to mimic reality. Afterwards, all the samples in the database are standardized, as this enhances the learning of the NNFs [129]. The database is then divided into a training set (with 80% of the database samples) and a test set (with the remaining database samples). The training set includes the ground truth labels for genuine and malicious vehicles, while the test set has no such information included. The NN_O and the NN_R are trained with their respective training sets.

In Fig. 6.2, ξ is plotted for the NN_R, NN_O, WNNF, and SNNF with a varying σ_{db}. Other parameter values are kept the same as highlighted earlier. We see that the WNNF has improved its performance compared to the NN_R but still performs below that of the NN_O. The same figure shows that the SNNF outperforms both the NN_R and NN_O.

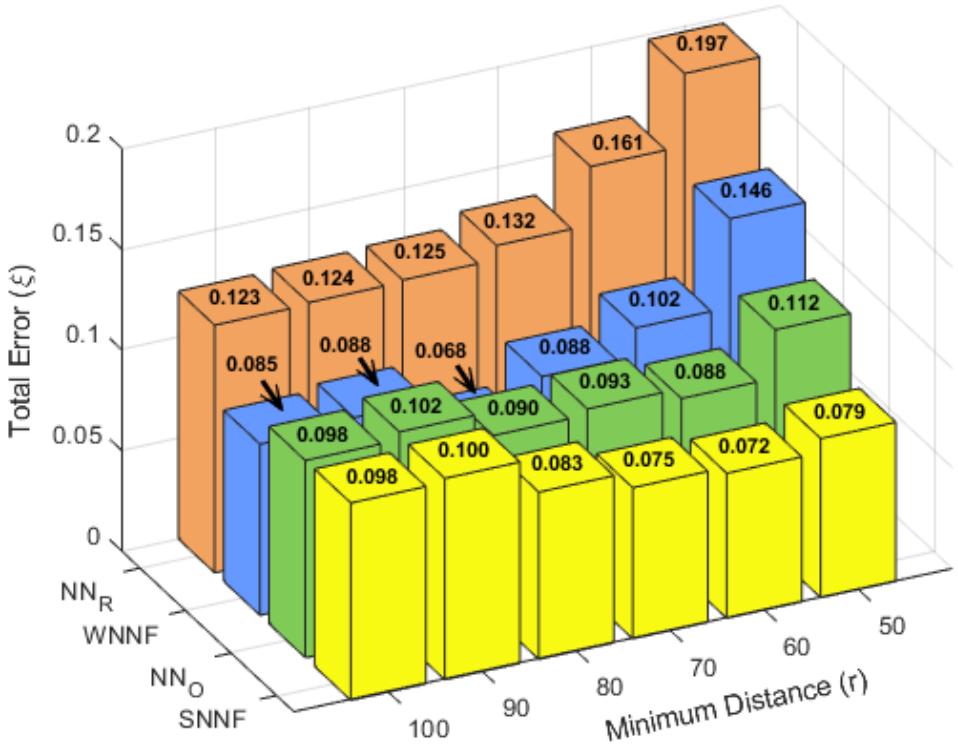

Figure 6.3: The performances of the NN_R, NN_O, WNNF, and SNNF with a changing minimum distance constraint for the malicious vehicle, *i.e.*, r. Here, we see an improved ξ for the SNNF compared to the NN_R, NN_O, and WNNF for $r \leq 70$ meters, but the WNNF performs best when $r > 70$ meters.

In Fig. 6.3, the performances of the NN_R, NN_O, WNNF, and SNNF are compared in a channel situation where the malicious vehicle varies its minimum distance, r, between its true and claimed locations. The values of the other parameters in this figure are $\sigma_{db} = 3$ dB, $\sigma_t = 2$ dB, and $\gamma = 3$. We see an interesting trend from the figure, *i.e.*, the SNNF performs efficiently in comparison to the NN_R, NN_O, and WNNF when r is less than or equal to 70 meters. However, when r increases beyond 70 meters, we see that the WNNF beats the performances of the NN_R, NN_O, SNNF.

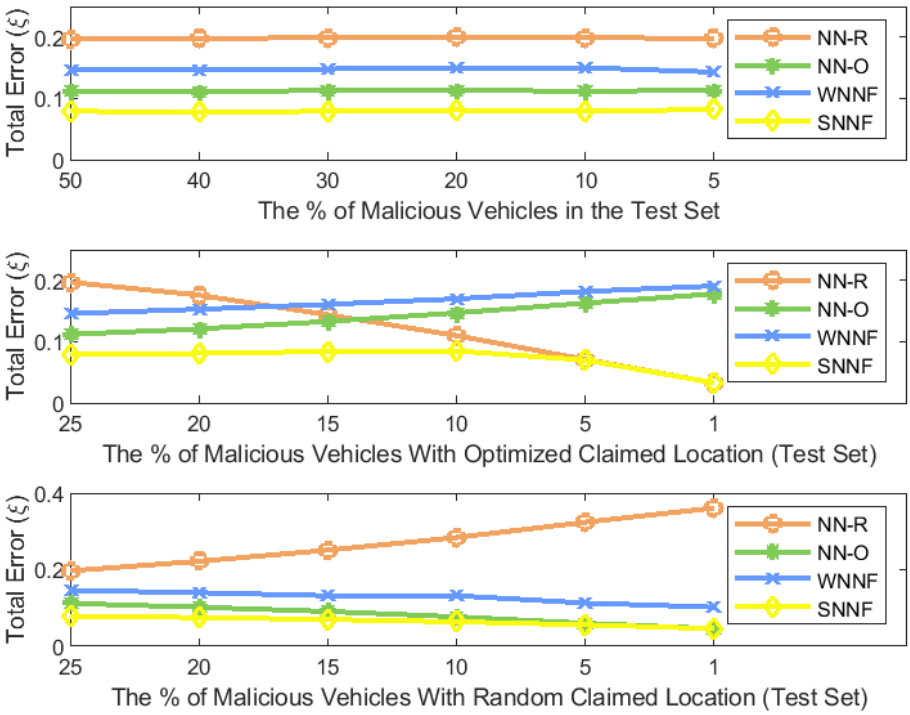

Figure 6.4: The performances of the NN_R, NN_O, WNNF, and SNNF in three different scenarios. First, the percentage of the malicious vehicles as a whole varies in the test set **(Top plot)**. Second, the percentage of the malicious vehicles, that optimize their claimed locations, changes in the test set only **(Middle plot)**. Third, the percentage of the malicious vehicles, that randomly claim their locations, varies in the test set only **(Bottom plot)**. In all three scenarios, we see that the SNNF is able to improve its performance.

In Fig. 6.4 (top plot), the performances of the NN_R, NN_O, WNNF, and SNNF are compared in a channel condition where the percentage of the malicious vehicles in the test set varies. Note that the proportions for the malicious vehicles that optimize their claimed locations and those that randomly claim their locations vary equally in the test set. The parameter values for σ_{db}, σ_t, γ, and r are set to 3 dB, 2 dB, 3, and 50 meters, respectively. We see that all the FNNs and NNFs are able to maintain their respective performances even when the percentage of the malicious vehicles in the test set reduces to low values. This claim is in line with our findings in Chapter 3 [28].

In the same figure (middle plot), an exciting scenario is covered. The values for all the channel parameters are kept the same as the top plot in Fig. 6.4. Here,

the percentage of the malicious vehicles that *optimize* their claimed locations varies in the test set only. We see that the performance of the NN_O deteriorates as the proportion of malicious vehicles that optimize their claimed location reduces. On the other hand, the SNNF is able to improve its performance.

Finally, in the same figure (bottom plot), a scenario similar to the middle plot is studied. None of the parameter values is changed. The only difference here is that the percentage of the malicious vehicles that *randomly* claim their locations now varies in the test set. Unlike the results in the middle plot, where the performance for the NN_R is improving, here the performance for the NN_R is deteriorating. However, we still see that the SNNF is building upon its performance as the percentage of the malicious vehicles (randomly claiming their locations) decreases to 1% in the test set.

6.5 Conclusion

This chapter combines standalone FNNs (each trained under different channel settings and threat models) to formulate a new combined weighted solution (termed as WNNF). The WNNF derives its parameters by statistically weighting, based on an input signal, standalone FNNs and uses these derived parameters to verify a vehicle's claimed location. A second framework (termed an SNNF) selects, based on the same input signal, the standalone neural FNN that is likely more suitable for verifying the vehicle's reported location. Both solutions are easy to deploy, do not require dynamic re-training in the field, and show improved performance relative to standalone FNN solutions. The new concepts introduced in this chapter are believed to be important in future wireless networks in the context of location verification, as well as other intrusion-detection scenarios.

Chapter 7

Conclusions and Future Work

This chapter concludes this thesis by briefly briefly summarizing its main contributions and pointing to potential future work in this area.

7.1 Thesis Conclusions

In Chapter 1, the significance of location acquisition and its verification was highlighted. A few of the notable works centred around location acquisition and verification were discussed. The operational gaps in the available LVSs were identified, explicitly stating how such LVSs faced operational difficulties when the channel conditions assumed at the time of their design did not exist. All such challenges made these LVSs impractical and unrealistic in real-world situations. Afterwards, the need to integrate NNs into the location verification arena was argued. The basics of NNs were also covered to educate the readers on how NNs function.

In Chapter 2, an NN-based location estimation framework was designed. This was followed by developing a systematic approach to formulate the internal architecture (especially the number of hidden layer neurons) of the framework. This approach was against the general practice of random search usually followed in formulating the internal architecture of the day-to-day NNs. Specifically, a CRB bound on the location accuracy was calculated and considered to intelligently decide the

number of neurons in the hidden layer of the proposed NN framework. This approach was validated through simulated as well as real-world RSS data. This approach is believed to ensure a less-time consuming and less likely over-fitting influenced NN solutions for numerous IoT applications beyond location estimation.

In Chapter 3, a few of the critical limitations that formal information-theoretic LVSs took into account for their operation were highlighted. One of such fundamental limitations identified for the information-theoretic LVSs was the *a-priori* information on the proportions of genuine and malicious users in the field. To address this limitation, an NN-based LVS was formulated. A comparison study for the two LVSs, *i.e.*, an information-theoretic LVS and the NN-based LVS, was carried out by considering simulated ToA measurements of the vehicles transmitted signals at multiple static BSs. It was shown how the working of the formulated NN-based LVS was free from any knowledge on the proportions of genuine and malicious users in the field. It was also demonstrated how the NN-based LVS exhibited a promising performance compared to the information-theoretic LVS in situations where the vehicles transmitted signals had random NLoS biases in them due to blockings from the numerous objects in the area.

In Chapter 4, the design of the formulated NN-based LVS was validated by considering real-world experimental data. The findings in this chapter were based on the RSS of the vehicles' transmitted signals measured at multiple static BSs. It was shown how the NN-based LVS was able to beat the performance of an information-theoretic LVS in scenarios where the adversary was (was not) able to optimize its attack location in an attempt to spoof the location verification framework.

A few of the crucial areas for the NN-based LVS solutions with a possibility for improved performance were targeted in the later chapters. These included the NN-based LVS training time, and the net classification performance gains.

In Chapter 5, an information-theoretic bound was utilized to monitor the training time of the NN-based LVS. This bound ensured a satisfactory performance by allowing a useful trade-off in learning-time *vs.* verification-performance for the NN-based LVS. The findings in Chapter 5 were based on the ToA measurements of the

vehicles transmitted signals.

In Chapter 6, new NNFs were formulated to enhance the location verification performance of the earlier designed NN-LVS solutions. A common problem with the NN-LVS solutions devised from Chapter 3 till Chapter 5 was their limited performance under given channel circumstances. Additionally, to avoid the danger of becoming obsolete and to keep themselves up to date to the variations in channel circumstances, periodical retraining (with updated training data) was required, something far from reality. To address these issues, two standalone FNNs trained under different channel conditions were combined to design two NNFs. One of the NNF followed a probabilistic approach for its operation, while the second NNF followed a selective approach for its working. Both the frameworks utilized the existing training of the standalone FNNs. Moreover, compared to FNNs, the new frameworks enhanced the overall classification performance significantly.

7.2 Future Research Directions

The research thus far can prove significant and can be projected further in the area of Simultaneous Reporting and Verification of Location (SRVL) in Beyond the Fifth-Generation[1] (B5G) technology networks. The mmWave-based communication is believed to be the key enabler for many applications in B5G. The SRVL is expected to integrate mmWave-based communication for its working, thus making it compatible with the other B5G applications.

The mmWave-based communication techniques can enhance SRVL in VANETs mainly due to the frequencies (high bandwidth) utilized. Also, the communication channels for mmWave are typically LoS. Both these issues aid SRVL. Multi-antenna techniques can be integrated into SRVL since a vehicle may be connected to only one

[1]Beyond Fifth-Generation (B5G) is an emerging technology that aims to support connectivity for a massive number of users and devices. This technology will offer data throughput rates in the multi-gigabits transmission range, will be flexible and intelligent to adapt to the changing protocols and topologies. In addition, it will offer support for intensive computation, storage applications, and top-end quality of service requirements to achieve ultra-high performances in various application areas.

BS in many scenarios. In such a situation, a single-antenna system cannot guarantee the reliability of the reported location information. This is because a malicious vehicle can modify its location metric (e.g. transmit power or transmission time) together with a false reported location to deceive an LVS [24].

The threat models that were considered in various chapters of this thesis followed some pre-defined parameters. In an attempt to spoof the system, a malicious user falsified its location following the threat model's parameters. In practical scenarios, we desire a parameter-free threat model that can update itself dynamically based on newly available data. A potential research area can be to develop intelligent LVS solutions that can address location spoofing attacks in parameter-free threat situations. The challenges in this context include the potential for over-fitting and identifying when to retrain the NN-based location verification framework.

Another promising direction to improve location verification is related to the integration (fusion) of many location-dependent services. For example, information partially dependent on location may come from other available reports within wireless vehicular networks such as broadcast messages, routing update messages, and radar reports. Although each of these additional components alone would not be useful for localization or location verification, such information may significantly improve location verification processes when analyzed collectively. Indeed, it has been previously shown how merging data from different sources makes user-authentication mechanisms more robust [130]. The NN solutions described in the previous chapters can be adapted to such merging of multi-source data through suitable training and adaptation.

Appendix A

Mathematics Behind a Neural Network for Classification

We derive a solution for $\frac{\partial \mathcal{L}_i}{\partial \mathbf{W}^{[2]}}$ in Eq. (1.24). From calculus, we know

$$\frac{\partial \mathcal{L}_i}{\partial \mathbf{W}^{[2]}} = \frac{\partial \mathcal{L}_i}{\partial a_i^{[2]}} \frac{\partial a_i^{[2]}}{\partial z_i^{[2]}} \frac{\partial z_i^{[2]}}{\partial \mathbf{W}^{[2]}}, \tag{A.1}$$

or,

$$\frac{\partial \mathcal{L}_i}{\partial \mathbf{W}^{[2]}} = \frac{\partial \mathcal{L}_i}{\partial z_i^{[2]}} \frac{\partial z_i^{[2]}}{\partial \mathbf{W}^{[2]}}. \tag{A.2}$$

Let $\nabla z_i^{[2]} = \frac{\partial \mathcal{L}_i}{\partial z_i^{[2]}}$, we can re-write Eq. (A.2) as below

$$\frac{\partial \mathcal{L}_i}{\partial \mathbf{W}^{[2]}} = \nabla z_i^{[2]} \frac{\partial z_i^{[2]}}{\partial \mathbf{W}^{[2]}}, \tag{A.3}$$

where $\frac{\partial z_i^{[2]}}{\partial \mathbf{W}^{[2]}}$ in Eq. (A.3) is solved as below

$$\frac{\partial z_i^{[2]}}{\partial \mathbf{W}^{[2]}} = \frac{\partial}{\partial \mathbf{W}^{[2]}} \Big(\mathbf{W}^{[2]} \, \mathbf{a}^{[1]} + b^{[2]} \Big), \tag{A.4}$$

After some simplification, we find

$$\frac{\partial z_i^{[2]}}{\partial \mathbf{W}^{[2]}} = \mathbf{a}^{[1]}. \tag{A.5}$$

Therefore, Eq. (A.3) becomes

$$\frac{\partial \mathcal{L}_i}{\partial \mathbf{W}^{[2]}} = \nabla z_i^{[2]}\, \mathbf{a}^{[1]}. \tag{A.6}$$

To find a solution to $\frac{\partial \mathcal{L}_i}{\partial b^{[2]}}$ in Eq. (1.25), we follow the chain rule of derivatives as below,

$$\frac{\partial \mathcal{L}_i}{\partial b^{[2]}} = \frac{\partial \mathcal{L}_i}{\partial a_i^{[2]}} \frac{\partial a_i^{[2]}}{\partial z_i^{[2]}} \frac{\partial z_i^{[2]}}{\partial b^{[2]}}. \tag{A.7}$$

We can also write Eq. (A.7) as

$$\frac{\partial \mathcal{L}_i}{\partial b^{[2]}} = \nabla z_i^{[2]} \frac{\partial z_i^{[2]}}{\partial b^{[2]}}, \tag{A.8}$$

where

$$\frac{\partial z_i^{[2]}}{\partial b^{[2]}} = \frac{\partial}{\partial b^{[2]}} \Big(\mathbf{W}^{[2]}\, \mathbf{a}^{[1]} + b^{[2]} \Big), \tag{A.9}$$

$$\frac{\partial z_i^{[2]}}{\partial b^{[2]}} = 1, \tag{A.10}$$

Hence, Eq. (A.8) becomes

$$\frac{\partial \mathcal{L}_i}{\partial b^{[2]}} = \nabla z_i^{[2]}. \tag{A.11}$$

We, next, solve for $\nabla z_i^{[2]}$ to further simplify Eqs. (A.6), and (A.11). We know

$$\nabla z_i^{[2]} = \frac{\partial \mathcal{L}_i}{\partial z_i^{[2]}} = \frac{\partial \mathcal{L}_i}{\partial a_i^{[2]}} \frac{\partial a_i^{[2]}}{\partial z_i^{[2]}}, \tag{A.12}$$

where

$$\begin{aligned} \frac{\partial \mathcal{L}_i}{\partial a_i^{[2]}} &= \frac{\partial}{\partial a_i^{[2]}} \Bigg[- \Big(y_i \log (a_i^{[2]}) + (1 - y_i) \log (1 - a_i^{[2]}) \Big) \Bigg], \\ &= -y_i \frac{\partial}{\partial a_i^{[2]}} \log (a_i^{[2]}) - (1 - y_i) \frac{\partial}{\partial a_i^{[2]}} \log (1 - a_i^{[2]}), \\ &= -\frac{y_i}{a_i^{[2]}} + \frac{1 - y_i}{1 - a_i^{[2]}}, \end{aligned} \tag{A.13}$$

After simplification,

$$\frac{\partial \mathcal{L}_i}{\partial a_i^{[2]}} = \frac{a_i^{[2]} - y_i}{a_i^{[2]} (1 - a_i^{[2]})}, \tag{A.14}$$

In Eq. (A.1), we solve $\frac{\partial a_i^{[2]}}{\partial z_i^{[2]}}$ as below

$$\begin{aligned}
\frac{\partial a_i^{[2]}}{\partial z_i^{[2]}} &= \frac{\partial}{\partial z_i^{[2]}}\left(\frac{1}{1+e^{-z_i^{[2]}}}\right), \\
&= \frac{\partial}{\partial z_i^{[2]}}\left(1+e^{-z_i^{[2]}}\right)^{-1}, \\
&= -\left(1+e^{-z_i^{[2]}}\right)^{-2}\frac{\partial}{\partial z_i^{[2]}}\left(1+e^{-z_i^{[2]}}\right), \\
&= \frac{e^{-z_i^{[2]}}}{\left(1+e^{-z_i^{[2]}}\right)^2}, \\
&= \frac{1}{1+e^{-z_i^{[2]}}}\left(\frac{1+e^{-z_i^{[2]}}-1}{1+e^{-z_i^{[2]}}}\right), \\
&= \frac{1}{1+e^{-z_i^{[2]}}}\left(\frac{1+e^{-z_i^{[2]}}}{1+e^{-z_i^{[2]}}}-\frac{1}{1+e^{-z_i^{[2]}}}\right), \\
&= \frac{1}{1+e^{-z_i^{[2]}}}\left(1-\frac{1}{1+e^{-z_i^{[2]}}}\right), \\
&= a_i^{[2]}(1-a_i^{[2]}).
\end{aligned} \tag{A.15}$$

Therefore,

$$\frac{\partial a_i^{[2]}}{\partial z_i^{[2]}} = a_i^{[2]}(1-a_i^{[2]}). \tag{A.16}$$

Substituting Equations (A.14), and (A.16) in Eq. (A.12),

$$\nabla z_i^{[2]} = \frac{a_i^{[2]}-y_i}{a_i^{[2]}\left(1-a_i^{[2]}\right)}\; a_i^{[2]}(1-a_i^{[2]}). \tag{A.17}$$

or

$$\nabla z_i^{[2]} = a_i^{[2]} - y_i. \tag{A.18}$$

Hence, we can finally write Eqs. (A.6), and (A.11) as below

$$\frac{\partial \mathcal{L}_i}{\partial \mathbf{W}^{[2]}} = \left(a_i^{[2]}-y_i\right)\mathbf{a}^{[1]}. \tag{A.19}$$

$$\frac{\partial \mathcal{L}_i}{\partial b^{[2]}} = a_i^{[2]} - y_i. \tag{A.20}$$

To solve for $\frac{\partial \mathcal{L}_i}{\partial \mathbf{W}^{[1]}}$ in Eq. (1.26), we follow the chain rule of derivatives as below

$$\frac{\partial \mathcal{L}_i}{\partial \mathbf{W}^{[1]}} = \frac{\partial \mathcal{L}_i}{\partial a_i^{[2]}} \frac{\partial a_i^{[2]}}{\partial z_i^{[2]}} \frac{\partial z_i^{[2]}}{\partial \mathbf{a}_i^{[1]}} \frac{\partial \mathbf{a}_i^{[1]}}{\partial \mathbf{z}_i^{[1]}} \frac{\partial \mathbf{z}_i^{[1]}}{\partial \mathbf{W}^{[1]}}. \tag{A.21}$$

Considering $\frac{\partial \mathcal{L}_i}{\partial a_i^{[2]}} \frac{\partial a_i^{[2]}}{\partial z_i^{[2]}} \frac{\partial z_i^{[2]}}{\partial \mathbf{a}_i^{[1]}} \frac{\partial \mathbf{a}_i^{[1]}}{\partial \mathbf{z}_i^{[1]}} = \frac{\partial \mathcal{L}_i}{\partial \mathbf{z}_i^{[1]}} = \nabla \mathbf{z}_i^{[1]}$, we re-write Eq. (A.21) as below

$$\frac{\partial \mathcal{L}_i}{\partial \mathbf{W}^{[1]}} = \nabla \mathbf{z}_i^{[1]} \frac{\partial \mathbf{z}_i^{[1]}}{\partial \mathbf{W}^{[1]}}, \tag{A.22}$$

where

$$\frac{\partial \mathbf{z}_i^{[1]}}{\partial \mathbf{W}^{[1]}} = \frac{\partial}{\partial \mathbf{W}^{[1]}} \Big(\mathbf{W}^{[1]} \mathbf{x} + \mathbf{b}^{[1]} \Big), \tag{A.23}$$

$$\frac{\partial \mathbf{z}_i^{[1]}}{\partial \mathbf{W}^{[1]}} = \mathbf{x}, \tag{A.24}$$

Therefore, we simplify (A.22) as

$$\frac{\partial \mathcal{L}_i}{\partial \mathbf{W}^{[1]}} = \nabla \mathbf{z}_i^{[1]} \mathbf{x}. \tag{A.25}$$

Finally, we solve for $\frac{\partial \mathcal{L}_i}{\partial \mathbf{b}^{[1]}}$ in Eq. (1.27) as below

$$\frac{\partial \mathcal{L}_i}{\partial \mathbf{b}^{[1]}} = \frac{\partial \mathcal{L}_i}{\partial a_i^{[2]}} \frac{\partial a_i^{[2]}}{\partial z_i^{[2]}} \frac{\partial z_i^{[2]}}{\partial \mathbf{a}_i^{[1]}} \frac{\partial \mathbf{a}_i^{[1]}}{\partial \mathbf{z}_i^{[1]}} \frac{\partial \mathbf{z}_i^{[1]}}{\partial \mathbf{b}^{[1]}}, \tag{A.26}$$

or

$$\frac{\partial \mathcal{L}_i}{\partial \mathbf{b}^{[1]}} = \nabla \mathbf{z}_i^{[1]} \frac{\partial \mathbf{z}_i^{[1]}}{\partial \mathbf{b}^{[1]}}, \tag{A.27}$$

where

$$\frac{\partial \mathbf{z}_i^{[1]}}{\partial \mathbf{b}^{[1]}} = \frac{\partial}{\partial \mathbf{b}^{[1]}} \Big(\mathbf{W}^{[1]} \mathbf{x} + \mathbf{b}^{[1]} \Big), \tag{A.28}$$

$$\frac{\partial \mathbf{z}_i^{[1]}}{\partial \mathbf{b}^{[1]}} = 1, \tag{A.29}$$

Therefore, Eq. (A.27) becomes

$$\frac{\partial \mathcal{L}_i}{\partial \mathbf{b}^{[1]}} = \nabla \mathbf{z}_i^{[1]}. \tag{A.30}$$

In order to further simplify the Equations (A.25), and (A.30), we solve for $\nabla \mathbf{z}_i^{[1]}$.

We know

$$\nabla \mathbf{z}_i^{[1]} = \frac{\partial \mathcal{L}_i}{\partial a_i^{[2]}} \frac{\partial a_i^{[2]}}{\partial z_i^{[2]}} \frac{\partial z_i^{[2]}}{\partial \mathbf{a}_i^{[1]}} \frac{\partial \mathbf{a}_i^{[1]}}{\partial \mathbf{z}_i^{[1]}}, \tag{A.31}$$

or

$$\nabla \mathbf{z}_i^{[1]} = \nabla z_i^{[2]} \frac{\partial z_i^{[2]}}{\partial \mathbf{a}_i^{[1]}} \frac{\partial \mathbf{a}_i^{[1]}}{\partial \mathbf{z}_i^{[1]}}, \tag{A.32}$$

where

$$\frac{\partial z_i^{[2]}}{\partial \mathbf{a}_i^{[1]}} = \frac{\partial}{\partial \mathbf{a}_i^{[1]}} \Big(\mathbf{W}^{[2]} \mathbf{a}_i^{[1]} + b^{[2]} \Big), \tag{A.33}$$

$$\frac{\partial z_i^{[2]}}{\partial \mathbf{a}_i^{[1]}} = \mathbf{W}^{[2]}, \tag{A.34}$$

and following Eq. (1.19), we have

$$\begin{aligned}
\frac{\partial \mathbf{a}_i^{[1]}}{\partial \mathbf{z}_i^{[1]}} &= \frac{\partial}{\partial \mathbf{z}_i^{[1]}} \psi(\mathbf{z}_i^{[1]}), \\
&= \frac{\partial}{\partial \mathbf{z}_i^{[1]}} \left(\frac{e^{\mathbf{z}_i^{[1]}} - e^{-\mathbf{z}_i^{[1]}}}{e^{\mathbf{z}_i^{[1]}} + e^{-\mathbf{z}_i^{[1]}}} \right), \\
&= \frac{\partial}{\partial \mathbf{z}_i^{[1]}} \Big(e^{\mathbf{z}_i^{[1]}} - e^{-\mathbf{z}_i^{[1]}} \Big) \Big(e^{\mathbf{z}_i^{[1]}} + e^{-\mathbf{z}_i^{[1]}} \Big)^{-1}, \\
&= \left(\Big(e^{\mathbf{z}_i^{[1]}} + e^{-\mathbf{z}_i^{[1]}} \Big)^{-1} \frac{\partial}{\partial \mathbf{z}_i^{[1]}} \Big(e^{\mathbf{z}_i^{[1]}} - e^{-\mathbf{z}_i^{[1]}} \Big) \right) + \left(\Big(e^{\mathbf{z}_i^{[1]}} - e^{-\mathbf{z}_i^{[1]}} \Big) \frac{\partial}{\partial \mathbf{z}_i^{[1]}} \Big(e^{\mathbf{z}_i^{[1]}} + e^{-\mathbf{z}_i^{[1]}} \Big)^{-1} \right), \\
&= \left(\Big(e^{\mathbf{z}_i^{[1]}} + e^{-\mathbf{z}_i^{[1]}} \Big)^{-1} \Big(e^{\mathbf{z}_i^{[1]}} + e^{-\mathbf{z}_i^{[1]}} \Big) \right) - \left(\Big(e^{\mathbf{z}_i^{[1]}} - e^{-\mathbf{z}_i^{[1]}} \Big)^2 \Big(e^{\mathbf{z}_i^{[1]}} + e^{-\mathbf{z}_i^{[1]}} \Big)^{-2} \right), \\
&= \frac{\Big(e^{\mathbf{z}_i^{[1]}} + e^{-\mathbf{z}_i^{[1]}} \Big)}{\Big(e^{\mathbf{z}_i^{[1]}} + e^{-\mathbf{z}_i^{[1]}} \Big)} - \frac{\Big(e^{\mathbf{z}_i^{[1]}} - e^{-\mathbf{z}_i^{[1]}} \Big)^2}{\Big(e^{\mathbf{z}_i^{[1]}} + e^{-\mathbf{z}_i^{[1]}} \Big)^2}, \\
&= 1 - \left(\frac{e^{\mathbf{z}_i^{[1]}} - e^{-\mathbf{z}_i^{[1]}}}{e^{\mathbf{z}_i^{[1]}} + e^{-\mathbf{z}_i^{[1]}}} \right)^2,
\end{aligned} \tag{A.35}$$

Or,

$$\frac{\partial \mathbf{a}_i^{[1]}}{\partial \mathbf{z}_i^{[1]}} = 1 - \Big(\psi(\mathbf{z}_i^{[1]}) \Big)^2. \tag{A.36}$$

Considering Equations (A.34), and (A.36), we can write Eq. (A.32) as below

$$\nabla \mathbf{z}_i^{[1]} = \nabla z_i^{[2]} \, \mathbf{W}^{[2]} \left(1 - \Big(\psi(\mathbf{z}_i^{[1]}) \Big)^2 \right). \tag{A.37}$$

We can thus write Equations (A.25) and (A.30) as below

$$\frac{\partial \mathcal{L}_i}{\partial \mathbf{W}^{[1]}} = \nabla z_i^{[2]} \, \mathbf{W}^{[2]} \left(1 - \left(\psi(\mathbf{z}_i^{[1]})\right)^2\right) \mathbf{x}. \tag{A.38}$$

$$\frac{\partial \mathcal{L}_i}{\partial \mathbf{b}^{[1]}} = \nabla z_i^{[2]} \, \mathbf{W}^{[2]} \left(1 - \left(\psi(\mathbf{z}_i^{[1]})\right)^2\right). \tag{A.39}$$

Appendix B

The Cramer-Rao Bound

The CRB represents a lower bound on the variance of an estimation. In other words, we can say that the CRB gives us a convenient way to characterize the best achievable performance or the lowest possible variance that an estimator can achieve.

We now derive the CRB for a likelihood function that corresponds to an unknown parameter q, *i.e.*, $f(\mathbf{y};q)$, where $\mathbf{y}$ is an M-dimensional observation vector and is given as $\mathbf{y} = [y_1, y_2, y_3, \ldots, y_M]^\top$. A likelihood function is the joint probability function with respect to the observation vector $\mathbf{y}$, parameterized by the unknown parameter q. We, therefore, must have

$$\int_{-\infty}^{\infty} f(\mathbf{y};q)\ d\mathbf{y} \ = \ 1. \tag{B.1}$$

Differentiate Eq. (B.1) with respect to the unknown parameter q, we get

$$\int_{-\infty}^{\infty} \frac{\partial}{\partial q}\ f(\mathbf{y};q)\ d\mathbf{y} \ = \ 0. \tag{B.2}$$

Rewriting Eq. (B.2) as below

$$\int_{-\infty}^{\infty} \frac{1}{f(\mathbf{y};q)} \left(\frac{\partial}{\partial q}\ f(\mathbf{y};q) \right) f(\mathbf{y};q)\ d\mathbf{y} \ = \ 0. \tag{B.3}$$

But $\frac{1}{P(\mathbf{y};q)} \frac{\partial}{\partial q}\ P(\mathbf{y};q) = \frac{\partial}{\partial q} \ln P(\mathbf{y};q)$, therefore, Eq. (B.3) becomes

$$\int_{-\infty}^{\infty} \left(\frac{\partial}{\partial q} \ln f(\mathbf{y};q) \right) f(\mathbf{y};q)\ d\mathbf{y} \ = \ 0. \tag{B.4}$$

Multiplying Eq.(B.4) by q on both sides, we get the equation below

$$\int_{-\infty}^{\infty} q \left(\frac{\partial}{\partial q} \ln f(\mathbf{y}; q) \right) f(\mathbf{y}; q)\, d\mathbf{y} \;=\; 0. \tag{B.5}$$

We, next, consider an unbiased estimator $\hat{q}$ of the parameter q, We know that $\mathbb{E}\{\hat{q}\} = q$, or

$$\int_{-\infty}^{\infty} \hat{q}\, f(\mathbf{y}; q)\, d\mathbf{y} \;=\; q. \tag{B.6}$$

Differentiating Eq. (B.6) with respect to q result in

$$\int_{-\infty}^{\infty} \hat{q}\, \frac{\partial}{\partial q}\, f(\mathbf{y}; q)\, d\mathbf{y} \;=\; 1. \tag{B.7}$$

$$\int_{-\infty}^{\infty} \hat{q}\, \frac{1}{f(\mathbf{y}; q)} \left(\frac{\partial}{\partial q}\, f(\mathbf{y}; q) \right) f(\mathbf{y}; q)\, d\mathbf{y} \;=\; 1. \tag{B.8}$$

$$\int_{-\infty}^{\infty} \hat{q} \left(\frac{\partial}{\partial q}\, \ln f(\mathbf{y}; q) \right) f(\mathbf{y}; q)\, d\mathbf{y} \;=\; 1. \tag{B.9}$$

Subtracting Eq. (B.5) from Eq.(B.9), and after some simplification, we get

$$\int_{-\infty}^{\infty} (\hat{q} - q) \left(\frac{\partial}{\partial q}\, \ln f(\mathbf{y}; q) \right) f(\mathbf{y}; q)\, d\mathbf{y} \;=\; 1, \tag{B.10}$$

where $(\hat{q} - q)$ is the estimation error. Thus Eq. (B.10) can be written as

$$\mathbb{E}\Big\{ (\hat{q} - q) \frac{\partial}{\partial q}\, \ln f(\mathbf{y}; q) \Big\} \;=\; 1. \tag{B.11}$$

From Cauchy-Schwarz inequality, we know $\mathbb{E}\{A^2\}\mathbb{E}\{B^2\} \geq \mathbb{E}^2\{AB\}$. We, therefore, can write Eq. (B.11) as

$$\mathbb{E}\Big\{ (\hat{q} - q)^2 \Big\} \mathbb{E}\Big\{ \Big(\frac{\partial}{\partial q}\, \ln f(\mathbf{y}; q) \Big)^2 \Big\} \geq \mathbb{E}^2\Big\{ (\hat{q} - q) \frac{\partial}{\partial q}\, \ln f(\mathbf{y}; q) \Big\} \;=\; (1)^2, \tag{B.12}$$

$$\mathbb{E}\Big\{ (\hat{q} - q)^2 \Big\} \mathbb{E}\Big\{ \Big(\frac{\partial}{\partial q}\, \ln f(\mathbf{y}; q) \Big)^2 \Big\} \geq 1, \tag{B.13}$$

$$\mathbb{E}\Big\{ (\hat{q} - q)^2 \Big\} \;\geq\; \frac{1}{\mathbb{E}\Big\{ \Big(\frac{\partial}{\partial q}\, \ln f(\mathbf{y}; q) \Big)^2 \Big\}}, \tag{B.14}$$

where $\mathbb{E}\{(\hat{q} - q)^2\}$ is the variance of the unbiased estimator and is referred to as

the CRB, and $\mathbb{E}\Big\{\Big(\frac{\partial}{\partial q}\ \ln f(\mathbf{y};q)\Big)^2\Big\}$ is the Fisher information, $\mathcal{F}$. This reveals an interesting result, *i.e.*, CRB is inversely proportional to $\mathcal{F}$. Elaborating this further, the larger the amount of information the log-likelihood function provides with respect to q, the larger is $\mathcal{F}$, and the lower is the estimation variance, *i.e.* the CRB.

B.0.1 CRB Example

This section is used to derive CRB for a sample case. We assume a BS that is estimating an unknown parameter q. The BS makes M measurements of the parameter q to be estimated. These measurements are given as, $\mathbf{y} = [y_1, y_2, y_3, \ldots, y_M]^\top$. Mathematically

$$y_k = q + v_k, \quad (k = 1,\ 2,\ 3, \ldots, M), \tag{B.15}$$

where v_k is the additive white Gaussian noise.

The liklihood function derived from $\mathbf{y}$ corresponding to the parameter q is given below

$$f(\mathbf{y};q) = \sqrt[M]{\frac{1}{2\pi\sigma^2}}\ e^{-\frac{1}{2\sigma^2}\sum_{k=1}^{M}(y_k-q)^2}, \tag{B.16}$$

$$\ln f(\mathbf{y};q) = \frac{1}{\sigma^2}\sum_{k=1}^{M}(y_k - q), \tag{B.17}$$

considering Eq. (B.15), we can rewrite Eq. (B.17) as below

$$\ln f(\mathbf{y};q) = \frac{1}{\sigma^2}\sum_{k=1}^{M} v_k. \tag{B.18}$$

From Eq. (B.14), CRB is the inverse of the Fisher information matrix, *i.e.*, CRB =

$\mathcal{F}^{-1} = \Big(\mathbb{E}\Big\{\Big(\frac{\partial}{\partial h}\ \ln f(\mathbf{y};q)\Big)^2\Big\}\Big)^{-1}$.

$$\begin{aligned}
\mathcal{F} &= \mathbb{E}\Big\{\Big(\frac{\partial}{\partial q}\ \ln f(\mathbf{y};q)\Big)^2\Big\},\\
&= \mathbb{E}\Big\{\Big(\frac{1}{\sigma^2}\sum_{k=1}^{M}\ v_k\Big)^2\Big\},\\
&= \frac{1}{\sigma^4}\mathbb{E}\Big\{\Big(\sum_{k=1}^{M}\ v_k\Big)^2\Big\},\\
&= \frac{1}{\sigma^4}\mathbb{E}\Big\{\Big(\sum_{k=1}^{M}\ v_k\Big)\ \Big(\sum_{\tilde{k}=1}^{M}\ v_{\tilde{k}}\Big)\Big\},\\
&= \frac{1}{\sigma^4}\mathbb{E}\Big\{\sum_{k=1}^{M}\sum_{\tilde{k}=1}^{M} v_k\ v_{\tilde{k}}\Big\},\\
&= \frac{1}{\sigma^4}\ \sum_{k=1}^{M}\sum_{\tilde{k}=1}^{M}\mathbb{E}\Big\{v_k\ v_{\tilde{k}}\Big\},
\end{aligned} \tag{B.19}$$

But v_k and $v_{\tilde{k}}$ are independent and identically Gaussian distributed, *i.e.*, $\mu = 0$, and variance σ^2. This means

$$f(x) = \begin{cases} \sigma^2, & \text{if } k = \tilde{k} \\ 0, & \text{if } k \neq \tilde{k} \end{cases} \tag{B.20}$$

Therefore

$$\mathbb{E}\Big\{v_k\ v_{\tilde{k}}\Big\} = \sigma^2\ \delta(k-\tilde{k}), \tag{B.21}$$

Hence

$$\begin{aligned}
\mathcal{F} &= \frac{1}{\sigma^4}\sum_{k=1}^{M}\sigma^2\ \delta(k-\tilde{k}),\\
&= \frac{1}{\sigma^4}\sum_{k=1}^{M}\sigma^2,\\
&= \frac{M}{\sigma^2}.
\end{aligned} \tag{B.22}$$

Finally

$$CRB = \frac{\sigma^2}{M}. \tag{B.23}$$

Appendix C

A Theoretical Lower Bound on Total Error

A quicker NN-LVS training stopping condition could be useful in that it can save training time, training data costs, and make for a faster NN-LVS. To achieve this, a new bound is introduced, referred to as the TLB. This bound, as will be shown, in conjunction with the usual learning process, can indeed result in a stopping of the NN-LVS training at an earlier time - a time when the NN-LVS is performing adequately. The TLB is calculated using information-theoretic analysis assuming *a priori* knowledge on the distribution of the bias term at each BS.

C.0.1 Likelihood Functions Under $\mathcal{H}_0$ and $\mathcal{H}_1$

The performance of the decision rule given in (5.11) is normally determined by the likelihood functions under $\mathcal{H}_0$ and $\mathcal{H}_1$. As such, in this subsection, the expressions of $f\left(y_i|\mathcal{H}_0\right)$ and $f\left(y_i|\mathcal{H}_1\right)$ are explicitly derived.

Proposition 1. *Following* (5.2)*, the likelihood function of* y_i *under* $\mathcal{H}_0$ *is derived as*

$$f(y_i|\mathcal{H}_0) = \frac{\rho_i}{2} Erfc\left(\frac{u_i + \rho_i\sigma_T^2 - y_i}{\sigma_T\sqrt{2}}\right) e^{\frac{\rho_i}{2}(2u_i+\rho_i\sigma_T^2-2y_i)}, \tag{C.1}$$

where $Erfc(\cdot)$ *is the complementary error function. Following* (5.6)*, the likelihood*

function of y_i under $\mathcal{H}_1$ is derived as

$$f(y_i|\mathcal{H}_1) = \frac{\rho_i}{2} Erfc\left(\frac{v_i + \rho_i\sigma_T^2 - y_i}{\sigma_T\sqrt{2}}\right) e^{\frac{\rho_i}{2}(2v_i+\rho_i\sigma_T^2-2y_i)}, \tag{C.2}$$

where $v_i = T_x + w_i$.

Proof. The proof of (C.1) is detailed in the following and the proof of (C.2) follows a similar procedure. Since u_i and ϕ_i are assumed to be independent to each other, following (5.2), the expression of $f(y_i|\mathcal{H}_0)$ is given by

$$f(y_i|\mathcal{H}_0) = \frac{\rho_i}{\sqrt{2\pi}\sigma_T} \int_0^\infty e^{-\rho_i t - \frac{(y_i-u_i-t)^2}{2\sigma_T^2}} dt. \tag{C.3}$$

With the aid of the following identity [131, Eq. (3.322.2)]

$$\int_0^\infty e^{-\frac{x^2}{4\beta}-\gamma x} dx = \sqrt{\pi\beta} e^{\beta\gamma^2} \text{Erfc}(\gamma\sqrt{\beta}), \text{ for } \beta > 0, \tag{C.4}$$

the integral in (C.3) is solved, which leads to the desired result in (C.1). This completes the proof of this proposition. □

Based on Proposition 1, the likelihood functions of $\mathbf{y}$ under $\mathcal{H}_0$ and $\mathcal{H}_1$ can be explicitly determined as per (5.5) and (5.8), respectively. Due to the complex expressions of these likelihood functions, the false positive rate and the detection rate (*i.e.*, α and β, respectively) cannot be derived in closed-form expressions. This leads to the fact that the Total Error ξ cannot be derived or numerically calculated, and can only be evaluated through lengthy Monte Carlo simulations. As such, in the following subsection, a lower bound on the Total Error ξ is determined, which can significantly facilitate the performance evaluation of the considered LVS.

C.0.2 A Lower Bound on Total Error

For the optimal binary decision rule, the minimum Total Error is given as [132, 133]

$$\xi^* = \frac{1}{2}\left(1 - \mathcal{V}_T(f(\mathbf{y}|\mathcal{H}_0), f(\mathbf{y}|\mathcal{H}_1))\right), \tag{C.5}$$

where $\mathcal{V}_T(f(\mathbf{y}|\mathcal{H}_0), f(\mathbf{y}|\mathcal{H}_1))$ is the total variation between $f(\mathbf{y}|\mathcal{H}_0)$ and $f(\mathbf{y}|\mathcal{H}_1)$. In general, computing $\mathcal{V}_T(f(\mathbf{y}|\mathcal{H}_0), f(\mathbf{y}|\mathcal{H}_1))$ occurs a high complexity, and thus Pinsker's inequality is normally adopted to upper bound it. Based on Pinsker's inequality

$$\mathcal{V}_T(f(\mathbf{y}|\mathcal{H}_1), f(\mathbf{y}|\mathcal{H}_0)) \leq \sqrt{\frac{1}{2}\mathcal{D}_{10}}, \tag{C.6}$$

or

$$\mathcal{V}_T(f(\mathbf{y}|\mathcal{H}_0), f(\mathbf{y}|\mathcal{H}_1)) \leq \sqrt{\frac{1}{2}\mathcal{D}_{01}}, \tag{C.7}$$

where $\mathcal{D}_{10}$ is the KL divergence from $f(\mathbf{y}|\mathcal{H}_1)$ to $f(\mathbf{y}|\mathcal{H}_0)$, which is given by

$$\mathcal{D}_{10} = \int_{\mathbf{y}} f(\mathbf{y}|\mathcal{H}_1) \log \frac{f(\mathbf{y}|\mathcal{H}_1)}{f(\mathbf{y}|\mathcal{H}_0)} d\mathbf{y}, \tag{C.8}$$

and $\mathcal{D}_{01}$ is the KL divergence from $f(\mathbf{y}|\mathcal{H}_0)$ to $f(\mathbf{y}|\mathcal{H}_1)$, which is given by

$$\mathcal{D}_{01} = \int_{\mathbf{y}} f(\mathbf{y}|\mathcal{H}_0) \log \frac{f(\mathbf{y}|\mathcal{H}_0)}{f(\mathbf{y}|\mathcal{H}_1)} d\mathbf{y}. \tag{C.9}$$

Both (C.6) and (C.7) are noted to be valid, although they are different due to the asymmetry of the KL divergence, which can be seen from (C.8) and (C.9). Following (C.5), (C.6), and (C.7), a lower bound on the minimum Total Error ξ^* can be written as

$$\xi^* \geq \xi_l \triangleq \frac{1}{2}\left(1 - \min\left[\sqrt{\frac{1}{2}\mathcal{D}_{10}}, \sqrt{\frac{1}{2}\mathcal{D}_{01}}\right]\right). \tag{C.10}$$

The lower bound ξ_l is of significant usefulness in the context of binary detection systems when the actual performance of a system cannot be directly examined. In the presence of bias, the first outcome achieved from this lower bound is on the determination of the optimal T_x (*i.e.*, T_x^*). From the malicious user-vehicle's point of view, maximizing the Total Error is the goal. As such, when non-zero bias terms are present we consider that T_x^* is the value that maximizes this lower bound ξ_l. We note that this lower bound ξ_l is based on perfect knowledge of some

system parameters (*i.e.*, the distribution of ϕ_i). However, this information may not be perfectly achieved in practical scenarios, which motivates us to adopt the NN methodology in the considered LVS. The usefulness of this lower bound on the minimum detection error probability lies in the fact that it can provide practical guidelines on when to stop training the machine.

C.0.3 Truncated Gaussian Distributed Bias

If the bias term ϕ_i follows a truncated normal distribution, the pdf of ϕ given in (5.4) should be updated to

$$f(\phi_i) = \frac{\sqrt{2}}{\sqrt{\pi}\sigma_i} e^{-\frac{\phi_i^2}{2\sigma_i^2}}, \tag{C.11}$$

where the mean and variance of the normal distribution before the truncation are zero and σ_i^2, respectively. Then, Proposition 1 should be updated to Proposition 2 as below:

Proposition 2. *Following* (5.2)*, the likelihood function of* y_i *under* $\mathcal{H}_0$ *is derived as*

$$f(y_i|\mathcal{H}_0) = \frac{e^{-\frac{(u_i-y_i)^2}{2(\sigma_i^2+\sigma_T^2)}}}{\sqrt{2\pi(\sigma_i^2+\sigma_T^2)}} Erfc\left(\frac{\sigma_i(u_i-y_i)}{\sigma_T\sqrt{2(\sigma_i^2+\sigma_T^2)}}\right). \tag{C.12}$$

Following (5.6)*, the likelihood function of* y_i *under* $\mathcal{H}_1$ *is derived as*

$$f(y_i|\mathcal{H}_1) = \frac{e^{-\frac{(v_i-y_i)^2}{2(\sigma_i^2+\sigma_T^2)}}}{\sqrt{2\pi(\sigma_i^2+\sigma_T^2)}} Erfc\left(\frac{\sigma_i(v_i-y_i)}{\sigma_T\sqrt{2(\sigma_i^2+\sigma_T^2)}}\right), \tag{C.13}$$

where $v_i = T_x + w_i$.

Proof. The proof of (C.12) is presented in the following, and the proof of (C.13) follows a similar procedure. Since u_i and ϕ_i are assumed to be independent of each

other, following (5.2), the expression of $f(y_i|\mathcal{H}_0)$ is given by

$$f(y_i|\mathcal{H}_0) = \frac{1}{2\pi\sigma_T\sigma_i}\int_0^\infty e^{-\frac{t^2}{2\sigma_i^2}-\frac{(y_i-u_i-t)^2}{2\sigma_T^2}}dt. \tag{C.14}$$

With the aid of the following identify [131, Eq. (3.322.2)]

$$\int_0^\infty e^{-\frac{x^2}{4\beta}-\gamma x}dx = \sqrt{\pi\beta}e^{\beta\gamma^2}\mathrm{Erfc}(\gamma\sqrt{\beta}),\ \text{for}\ \beta > 0, \tag{C.15}$$

the integral in (C.14) is solved, which leads to the desired result in (C.12). This completes the proof of this proposition. □

Appendix D

Estimation of P_n

In estimating P_n, we would like to ensure that any loss in the network performance (distance accuracy) introduced by constraining that number is not too large. As we now show, the Universal Approximation Theorem (UAT) [134] can allow insight into that. Loosely speaking, the UAT states that there always exists a NN that can approximate any input function, $f(x)$, with any output function, $g(x)$, to any arbitrary accuracy ε, i.e. $|g(x) - f(x)| < \varepsilon$. In our case $x = \text{RSS}$ and $f(x)$ maps to the distance, d, between transmitter and receiver.

To make progress let us consider a single hidden layer NN with transfer functions of the sigmoid form (our result will be independent of this choice). A neuron is modelled by $\zeta(\omega x + b)$, where $\zeta(z) \equiv 1/(1 + e^{-z})$. It is straightforward to show that values of ω and b can be chosen to 'force' the transfer function into a step form where the step occurs at $-b/\omega$. Further, by using P_n transfer functions (P_n neurons) connected in a single layer network, it is straightforward to show that a series of rectangles can be formed at the output [135]. That is, you can create a network that can model any input function $f(x)$ as an output function $g(x)$ consisting of a series of P_n rectangles. If we simplify this further and make all the rectangles of equal width, we can easily determine P_n such that $|g(x) - f(x)| < \varepsilon$, where ε now represents the standard deviation in the distance difference between the two functions. The CRB on the distance estimate for log-normal shadowing is $\ln(10)\sigma_{dB}d/(10n)$. Therefore, a good estimate of our required P_n would be one that ensures $\ln(10)\sigma_{dB}d/(10n) > \varepsilon$.

References

[1] T. Leinmuller, E. Schoch, F. Kargl, and C. Maihofer, "Influence of falsified position data on geographic ad-hoc routing," in *Proceedings of the European Workshop on Security in Ad-hoc and Sensor Networks*, Visegrad, Hungary, Jul. 2005, pp. 102–112.

[2] T. Leinmuller and E. Schoch, "Greedy routing in highway scenarios: The impact of position faking nodes," in *Proceedings of the Workshop On Intelligent Transportation*, Jan. 2006, pp. 1–6.

[3] M. Rabayah and R. Malaney, "A new scalable hybrid routing protocol for VANETs," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 6, pp. 2625–2635, Jul. 2012.

[4] S. Chen, Y. Zhang, and W. Trappe, "Inverting sensor networks and actuating the environment for spatio-temporal access control," in *Proceedings of the ACM workshop on Security of ad hoc and sensor networks*, Alexandria, VA, USA, Oct. 2006, pp. 1–12.

[5] S. Capkun, M. Cagalj, G. Karame, and N. Tippenhauer, "Integrity regions: Authentication through presence in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 11, pp. 1608–1621, Nov. 2010.

[6] M. Ayaida, H. Fouchal, L. Afilal, and Y. Ghamri, "A comparison of reactive, grid and hierarchical location-based services for VANETs," in *Proceedings of the IEEE Vehicular Technology Conference (VTC)*, Quebec City, QC, Canada, Sep. 2012, pp. 1–5.

[7] L. Chen *et al.*, "Robustness, security and privacy in location-based services for future IoT: A survey," *IEEE Access*, vol. 5, pp. 8956–8977, Apr. 2017.

[8] R. Malaney, "A secure and energy efficient scheme for wireless voip emergency service.," in *Proceedings of the IEEE Global Communications Conference (GlobeCOM)*, San Francisco, CA, USA, Nov. 2006, pp. 1–6.

[9] M. Damiani, E. Bertino, B. Catania, and P. Perlasca, "GEO-RBAC: A spatially aware RBAC," *ACM Transactions on Information and System Security (TISSEC)*, vol. 10, no. 1, pp. 1–42, Feb. 2007.

[10] M. Raya and J. Hubaux, "The security of vehicular ad hoc networks," in *Proceedings of the ACM workshop on Security of ad hoc and sensor networks*, Alexandria, VA, USA, Nov. 2005, pp. 11–21.

[11] P. Papadimitratos, V. Gligor, and J. Hubaux, "Securing vehicular communications - assumptions, requirements, and principles," in *Proceedings of the Embedded Security in Cars (ESCAR)*, Berlin, Germany, Nov. 2006, pp. 5–14.

[12] M. Raya and J. Hubaux, "Securing vehicular ad-hoc networks," *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, Jul. 2007.

[13] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, and J. Freudiger, "Secure vehicular communication systems: Design and architecture," *IEEE Communications Magazine*, vol. 46, no. 11, Nov. 2008.

[14] W. Jabbar and R. Malaney, "Mobility models and the performance of location-based routing in VANETs," in *Proceedings of the IEEE Vehicular Technology Conference (VTC)*, Victoria, BC, Canada, Nov.–Dec. 2020, pp. 1–5.

[15] F. Knorr, D. Baselt, M. Schreckenberg, and M. Mauve, "Reducing traffic jams via VANETs," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 8, pp. 3490–3498, Jul. 2012.

[16] C. Barba, M. Mateos, P. Soto, A. Mezher, and M. Igartua, "Smart city for VANETs using warning messages, traffic statistics and intelligent traffic lights," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, Madrid, Spain, Jun. 2012, pp. 902–907.

[17] "Global status report on road safety 2018." Accessed on: July 28, 2021. (Jun. 2018), [Online]. Available: `https://www.who.int/publications/i/item/9789241565684`.

[18] "Who global status report on road safety 2013: Supporting a decade of action." Accessed on: July 28, 2021. (Apr. 2013), [Online]. Available: `https://apps.who.int/iris/handle/10665/78256`.

[19] S. Yan, R. Malaney, I. Nevat, and G. Peters, "An information theoretic location verification system for wireless networks," in *Proceedings of the IEEE Global Communications Conference (GlobeCOM)*, Dec. 2012, pp. 5415–5420.

[20] A. Jaeger, N. Bißmeyer, H. Stübing, and S. Huss, "A novel framework for efficient mobility data verification in vehicular ad-hoc networks," *International Journal of Intelligent Transportation Systems Research*, vol. 10, no. 1, pp. 11–21, Jan. 2012.

[21] B. Yu, C. Xu, and B. Xiao, "Detecting sybil attacks in VANETs," *Journal of Parallel and Distributed Computing*, vol. 73, no. 6, pp. 746–756, Jun. 2013.

[22] S. Yan and R. Malaney, "Location verification systems in emerging wireless networks," *ZTE Communications*, vol. 11, no. 3, pp. 03–10, Jul. 2013.

[23] S. Yan, R. Malaney, I. Nevat, and G. Peters, "Timing information in wireless communications and optimal location verification frameworks," in *Proceedings of the IEEE Australian Communications Theory Workshop (AusCTW)*, Sydney, Australia, Feb. 2014, pp. 144–149.

[24] S. Yan, I. Nevat, G. Peters, and R. Malaney, "Location verification systems under spatially correlated shadowing," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 4132–4144, Jun. 2016.

[25] S. Yan, G. Peters, I. Nevat, and R. Malaney, "Location verification systems based on received signal strength with unknown transmit power," *IEEE Communications Letters*, vol. 22, no. 3, pp. 650–653, Mar. 2018.

[26] P. Monteiro, J. Rebelatto, and R. Souza, "Information-theoretic location verification system with directional antennas for vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 93–103, Jan. 2016.

[27] D. Sheet, O. Kaiwartya, A. Abdullah, Y. Cao, and H. Ahmed, "Location information verification using transferable belief model for geographic routing in vehicular ad hoc networks," *IET Intelligent Transportation Systems*, vol. 11, no. 2, pp. 53–60, Mar. 2017.

[28] U. Ihsan, Z. Wang, R. Malaney, A. Dempster, and S. Yan, "Artificial intelligence and location verification in vehicular networks," in *Proceedings of the IEEE Global Communications Conference (GlobeCOM)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.

[29] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global positioning system: theory and practice*. Springer Science & Business Media, 2012.

[30] S. Lawrence, C. Giles, A. Tsoi, and A. Back, "Face recognition: A convolutional neural-network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, Jan. 1997.

[31] E. Abdulhay, N. Arunkumar, K. Narasimhan, E. Vellaiappan, and V. Venkatraman, "Gait and tremor investigation using machine learning techniques for the diagnosis of parkinson disease," *Future Generation Computer Systems*, vol. 83, pp. 366–373, Jun. 2018.

[32] S. Khan, H. Rahmani, S. Shah, and M. Bennamoun, "A guide to convolutional neural networks for computer vision," *Synthesis Lectures on Computer Vision*, vol. 8, no. 1, pp. 1–207, Feb. 2018.

[33] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May. 2015.

[34] S. Irtza, V. Sethu, E. Ambikairajah, and H. Li, "End-to-end hierarchical language identification system," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, AB, Canada, Apr. 2018, pp. 5199–5203.

[35] P. Matejka, O. Glembek, O. Novotny, O. Plchot, and F. Grezl, "Analysis of DNN approaches to speaker identification," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, Mar. 2016, pp. 5100–5104.

[36] Y. Costa, L. Oliveira, and C. Silla, "An evaluation of convolutional neural networks for music classification using spectrograms," *Applied Soft Computing*, vol. 52, pp. 28–38, Mar. 2017.

[37] A. Graves, N. Jaitly, and A. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, Olomouc, Czech Republic, Dec. 2013, pp. 273–278.

[38] L. Fridman *et al.*, "MIT autonomous vehicle technology study: Large-scale deep learning based analysis of driver behavior and interaction with automation," *arXiv preprint arXiv:1711.06976*, vol. 1, Nov. 2017.

[39] J. Werb and C. Lanzl, "Designing a positioning system for finding things and people indoors," *IEEE spectrum*, vol. 35, no. 9, pp. 71–78, Sep. 1998.

[40] M. Kim, E. Lee, and Y. Lee, "Simulation of intersection collision avoidance system in wireless sensor networks," in *Proceedings of the IEEE International Symposium on Geoscience and Remote Sensing*, Denver, CO, USA, Aug. 2006, pp. 2876–2879.

[41] A. Boukerche, H. Oliveira, E. Nakamura, and A. Loureiro, "Vehicular ad hoc networks: A new challenge for localization-based systems," *Computer Communications*, vol. 31, no. 12, pp. 2838–2849, Jul. 2008.

[42] S. Savasta, M. Pini, and G. Marfia, "Performance assessment of a commercial GPS receiver for networking applications," in *Proceedings of the IEEE Consumer Communications and Networking Conference*, Las Vegas, NV, USA, Feb. 2008, pp. 613–617.

[43] G. Morgan and G. Johnston, "Differential GPS positioning," *Electronics & Communication Engineering Journal*, vol. 7, no. 1, pp. 11–21, Feb. 1995.

[44] T. King, H. Fubler, M. Transier, and W. Effelsberg, "Dead-reckoning for position-based forwarding on highways," in *Proceedings of the ACM International Workshop on Intelligent Transportation (WIT)*, Hamburg, Germany, Mar. 2006, pp. 199–204.

[45] E. Lee, S. Yang, S. Oh, and M. Gerla, "RF-GPS: RFID assisted localization in VANETs," in *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems*, Macau, China, Nov. 2009, pp. 621–626.

[46] S. Bauer, M. Obst, R. Streiter, and G. Wanielik, "Evaluation of shadow maps for non-line-of-sight detection in urban GNSS vehicle localization with VANETs-The GAIN approach," in *Proceedings of the IEEE Vehicular Technology Conference (VTC Spring)*, Dresden, Germany, Jun. 2013, pp. 1–5.

[47] K. Golestan, S. Seifzadeh, M. Kamel, F. Karray, and F. Sattar, "Vehicle localization in VANETs using data fusion and V2V communication," in *Proceedings of the ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*, Paphos, Cyprus, Oct. 2012, pp. 123–130.

[48] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, Mar. 2002.

[49] G. Hoangt, B. Denis, J. Hairri, and D. Slock, "Cooperative localization in VANETS: An experimental proof-of-concept combining GPS, IR-UWB ranging and V2V communications," in *Proceedings of the IEEE Workshop on Positioning, Navigation and Communications (WPNC)*, Bremen, Germany, Dec. 2018, pp. 1–6.

[50] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, no. 5, pp. 28–34, Oct. 2000.

[51] L. Girod and D. Estrin, "Robust range estimation using acoustic and multimodal sensing," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, Maui, HI, USA, Nov. 2001, pp. 1312–1320.

[52] D. Nicules and B. Nath, "Ad-hoc position system (APS) using AoA," in *Proceedings of the IEEE INFOCOM*, San Francisco, CA, USA, Jul. 2003, pp. 1–10.

[53] R. Nagpal *et al.*, "Organizing a global coordinate system from local information on an amorphous computer," Aug. 1999.

[54] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *Telecommunication Systems*, vol. 22, no. 1, pp. 267–280, Jan. 2003.

[55] H. Zou, X. Lu, H. Jiang, and L. Xie, "A fast and precise indoor localization algorithm based on an online sequential extreme learning machine," *Sensors*, vol. 15, no. 1, pp. 1804–1824, Jan. 2015.

[56] G. Felix, M. Siller, and E. Alvarez, "A fingerprinting indoor localization algorithm based deep learning," in *Proceedings of the 8th IEEE International Conference on Ubiquitous and Future Networks (ICUFN)*, Vienna, Austria, Jul. 2016, pp. 1006–1011.

[57] Y. Robinson, S. Vimal, E. Julie, K. Narayanan, and S. Rho, "3-dimensional manifold and machine learning based localization algorithm for wireless sensor networks," *Wireless Personal Communications*, pp. 1–19, Mar. 2021.

[58] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," in *Proceedings of the ACM Workshop on Wireless Security*, New York, NY, United States, Sep. 2003, pp. 1–10.

[59] D. Singelee and B. Preneel, "Location verification using secure distance bounding protocols," in *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems*, Washington, DC, USA, Nov. 2005, pp. 1–7.

[60] S. Capkun and J. Hubaux, "Secure positioning in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 221–232, Feb. 2006.

[61] T. Leinmuller, E. Schoch, and F. Kargl, "Position verification approaches for vehicular ad hoc networks," *IEEE Wireless Communications*, vol. 13, no. 5, pp. 16–21, Oct. 2006.

[62] K. Liu, N. Abu, and K. Kang, "Location verification and trust management for resilient geographic routing," *Journal of parallel and distributed computing*, vol. 67, no. 2, pp. 215–228, Oct. 2006.

[63] C. Harsch, A. Festag, and P. Papadimitratos, "Secure position-based routing for VANETs," in *Proceedings of the IEEE Vehicular Technology Conference (VTC)*, Baltimore, MD, USA, Oct. 2007, pp. 26–30.

[64] S. Capkun, K. Rasmussen, M. Cagalj, and M. Srivastava, "Secure location verification with hidden and mobile base stations," *IEEE Transactions on Mobile Computing*, vol. 7, no. 4, pp. 470–483, Feb. 2008.

[65] G. Yan, S. Olariu, and M. Weigle, "Providing VANET security through active position detection," *Computer Communications*, vol. 31, no. 12, pp. 2883–2897, Jul. 2008.

[66] O. Abumansoor and A. Boukerche, "A secure cooperative approach for nonline-of-sight location verification in vanet," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 275–285, Nov. 2011.

[67] J. Chiang, J. Haas, J. Choi, and Y. Hu, "Secure location verification using simultaneous multilateration," *IEEE Transactions on Wireless Communications*, vol. 11, no. 2, pp. 584–591, Dec. 2011.

[68] Y. Wei and Y. Guan, "Lightweight location verification algorithms for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 5, pp. 938–950, Jan. 2012.

[69] P. Zhang, Z. Zhang, and A. Boukerche, "Cooperative location verification for vehicular ad-hoc networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*, Ottawa, ON, Canada, Jun. 2012, pp. 37–41.

[70] F. Malandrino, C. Casetti, C. Chiasserini, M. Fiore, and R. Yokoyama, "A-VIP: Anonymous verification and inference of positions in vehicular networks," in *Proceedings of the IEEE INFOCOM*, Turin, Italy, Apr. 2013, pp. 105–109.

[71] C. Palazzi, S. Ferretti, M. Roccetti, G. Pau, and M. Gerla, "How do you quickly choreograph inter-vehicular communications? A fast vehicle-to-vehicle multi-hop broadcast algorithm, explained," in *Proceedings of the IEEE Consumer Communications and Networking Conference*, Las Vegas, NV, USA, Jan. 2007, pp. 960–964.

[72] W. Jaballah, M. Conti, M. Mosbah, and C. Palazzi, "Secure verification of location claims on a vehicular safety application," in *Proceedings of the International Conference on Computer Communication and Networks (ICCCN)*, Nassau, Bahamas, Aug. 2013, pp. 1–7.

[73] R. Kasana, K. Sushil, O. Kaiwartya, W. Yan, and Y. Cao, "Location error resilient geographical routing for vehicular ad-hoc networks," *IET Intelligent Transport Systems*, vol. 11, no. 8, pp. 450–458, Oct. 2017.

[74] I. Kim, B. Kim, and J. Song, "An efficient location verification scheme for static wireless sensor networks," *Sensors*, vol. 17, no. 2, p. 225, Jan. 2017.

[75] G. Caparra, M. Centenaro, N. Laurenti, and S. Tomasin, "Optimization of anchor nodes' usage for location verification systems," in *Proceedings of the International Conference on Localization and GNSS (ICL-GNSS)*, Nottingham, UK, Jun. 2017, pp. 1–6.

[76] C. Vaas, M. Juuti, N. Asokan, and I. Martinovic, "Get in line: Ongoing co-presence verification of a vehicle formation based on driving trajectories," in *Proceedings of the IEEE European Symposium on Security and Privacy (Euro S&P)*, Apr. 2018, pp. 199–213.

[77] A. Dua, N. Kumar, A. Kumar, and W. Susilo, "Secure message communication protocol among vehicles in smart city," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4359–4373, May 2018.

[78] X. Liu, S. Su, F. Han, Y. Liu, and Z. Pan, "A range-based secure localization algorithm for wireless sensor networks," *IEEE Sensors Journal*, vol. 19, no. 2, pp. 785–796, Oct. 2018.

[79] S. Yan, R. Malaney, I. Nevat, and G. Peters, "Optimal information-theoretic wireless location verification," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 7, pp. 3410–3422, Sep. 2014.

[80] M. Monteiro, J. Rebelatto, and R. Souza, "Information-theoretic location verification system with directional antennas," in *Proceedings of the IEEE International Telecommunications Symposium (ITS)*, Sao Paulo, Brazil, Aug. 2014, pp. 1–5.

[81] S. Yan, R. Malaney, I. Nevat, and G. Peters, "Location verification systems for VANETs in Rician fading channels," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 7, pp. 5652–5664, Jul. 2016.

[82] S. Yan, R. Malaney, I. Nevat, and G. W. Peters, "Location spoofing detection for vanets by a single base station in rician fading channels," in *Proceedings*

of the IEEE Vehicular Technology Conference (VTC), Glasgow, UK, May. 2015, pp. 1–6.

[83] R. Gholami and G. Hodtani, "A more general information theoretic study of wireless location verification system," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9938–9950, Jun. 2020.

[84] A. Brighente, F. Formaggio, M. Centenaro, G. M. Di Nunzio, and S. Tomasin, "Location-verification and network planning via machine learning approaches," in *Proceedings of the IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*, Avignon, France, Jun. 2019, pp. 1–7.

[85] A. Brighente, F. Formaggio, G. Ruvoletto, and S. Tomasin, "Ranking-based attacks to in-region location verification systems," in *Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS)*, IEEE, Delft, Netherlands, Dec. 2019, pp. 1–6.

[86] S. Tomasin, A. Brighente, F. Formaggio, and G. Ruvoletto, "Physical-layer location verification by machine learning," *Machine Learning for Future Wireless Communications*, pp. 425–438, Jan. 2020.

[87] T. Elsken, H. Metzen, and F. Hutter, "Neural architecture search: A survey," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 1997–2017, Mar. 2019.

[88] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, `http://www.deeplearningbook.org`.

[89] N. Alam, A. Balaei, and A. Dempster, "Relative positioning enhancement in VANETs: A tight integration approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 47–55, Jul. 2012.

[90] G. Hoang, B. Denis, J. Harri, and D. Slock, "Cooperative localization in GNSS-aided VANETs with accurate IR-UWB range measurements," in *Proceedings of the IEEE Workshop on Positioning, Navigation and Communications*, Bremen, Germany, Oct. 2016, pp. 1–6.

[91] S. Cruz, T. Abrudan, Z. Xiao, N. Trigoni, and J. Barros, "Neighbor-aided localization in vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2693–2702, Feb. 2017.

[92] H. Chen, Y. Zhang, W. Li, X. Tao, and P. Zhang, "ConFi: Convolutional neural networks based indoor Wi-Fi localization using channel state information," *IEEE Access*, vol. 5, pp. 18 066–18 074, Sep. 2017.

[93] A. Kumar and V. Jain, "Feed forward neural network-based sensor node localization in Internet of Things," in *Progress in Computing, Analytics and Networking*, Springer, 2018, pp. 795–804.

[94] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. 10, pp. 281–305, Feb. 2012.

[95] J. Heaton, *Artificial Intelligence for Humans, Volume 3: Deep Learning and Neural Networks, The Science of Microfabrication*. Heaton Research, Inc., Dec. 2015.

[96] R. Malaney, "Securing Wi-Fi networks with position verification: Extended version," *International Journal of Security and Networks*, vol. 2, no. 1-2, pp. 27–36, 2007.

[97] U. Ihsan, R. Malaney, and S. Yan, "Machine learning and location verification in vehicular networks," in *Proceedings of the 8th IEEE/CIC International Conference on Communications in China (ICCC2019)*, Changchun, China, Aug. 2019, pp. 91–95.

[98] U. Ihsan, S. Yan, and R. Malaney, "Location verification for emerging wireless vehicular networks," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 261–10 272, Aug. 2019.

[99] T. Nakama, "Comparisons of single-and multiple-hidden-layer neural networks," in *Proceedings of the International Symposium on Neural Networks*, Guilin, China, Jun. 2011, pp. 270–279.

[100] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the Neural Information Processing Systems*, Lake Tahoe, NV, USA, Jan. 2012, pp. 1097–1105.

[101] C. Bircanoglu and N. Arica, "A comparison of activation functions in artificial neural networks," in *Proceedings of the IEEE Signal Processing and Communications Applications Conference (SIU)*, Izmir, Turkey, May, 2018, pp. 1–4.

[102] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad-hoc networks," *IEEE network*, vol. 15, no. 6, pp. 30–39, Dec. 2001.

[103] C. Steffes, R. Kaune, S. Rau, and F. Fkie, "Determining Times of Arrival of transponder signals in a sensor network using GPS time synchronization," *Jahrestagung der Gesellschaft für Informatik, Berlin*, Oct. 2011.

[104] I. Martin, F. Barcelo, and E. Zola, "Software based measurement of round trip time observables for location in IEEE 802.11 networks," in *Proceedings of the IEEE International Conference on Telecommunications (ConTEL)*, Zagreb, Croatia, Jun. 2013, pp. 95–102.

[105] J. Neyman and E. Pearson, "IX. On the problem of the most efficient tests of statistical hypotheses," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 231, no. 694-706, pp. 289–337, Feb. 1933.

[106] H. Hartenstein and L. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Communications Magazine*, vol. 46, no. 6, pp. 164–171, Jun. 2008.

[107] S. Kullback and R. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, Mar. 1951.

[108] S. Eguchi and J. Copas, "Interpreting Kullback–Leibler divergence with the Neyman–Pearson lemma," *Journal of Multivariate Analysis*, vol. 97, no. 9, pp. 2034–2040, Oct. 2006.

[109] R. Meneguette, E. Robson, and A. Loureiro, *Intelligent Transport System in Smart Cities: Aspects and Challenges of Vehicular Networks and Cloud*, 1st ed. Springer, May. 2018.

[110] B. Ramsundar *et al.*, "Massively multitask networks for drug discovery," *arXiv:1502.02072*, Feb. 2015.

[111] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proccedings of the European Conference on Computer Vision*, Springer, Graz, Austria, Jul. 2006, pp. 430–443.

[112] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[113] A. Giusti *et al.*, "A machine learning approach to visual perception of forest trails for mobile robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 661–667, Jul. 2016.

[114] E. Guizzo, "How Google's self-driving car works," *IEEE Spectrum*, vol. 18, no. 7, pp. 1132–1141, Oct. 2011.

[115] N. Kumar, S. Zeadally, and J. Rodrigues, "Vehicular delay-tolerant networks for smart grid data management using mobile edge computing," *IEEE Communications Magazine*, vol. 54, no. 10, pp. 60–66, Oct. 2016.

[116] A. Dua, N. Kumar, and S. Bawa, "Game theoretic approach for real-time data dissemination and offloading in vehicular ad hoc networks," *Journal of Real-Time Image Processing*, vol. 13, no. 3, pp. 627–644, Sep. 2017.

[117] O. Khan, M. Shah, I. Din, B. Kim, and H. Khattak, "Leveraging named data networking for fragmented networks in smart metropolitan cities," *IEEE Access*, vol. 6, pp. 75 899–75 911, Nov. 2018.

[118] I. Din, B. Kim, S. Hassan, M. Guizani, and M. Atiquzzaman, "Information-centric network-based vehicular communications: Overview and research opportunities," *Sensors*, vol. 18, no. 11, Nov. 2018.

[119] I. Din, H. Asmat, and M. Guizani, "A review of information centric network-based internet of things: Communication architectures, design issues, and

research opportunities," *Multimedia Tools and Applications*, pp. 1–16, Dec. 2018.

[120] T. Cover and T. Joy, *Elements of information theory*. John Wiley & Sons, Apr. 2005.

[121] M. Barkat, "Signal detection and estimation," in 2nd ed. Boston: Artech House, Dec. 1991, ch. 5.

[122] R. Barzegar and A. Moghaddam, "Combining the advantages of neural networks using the concept of committee machine in the groundwater salinity prediction," *Modeling Earth Systems and Environment*, vol. 2, no. 1, p. 26, Jan. 2016.

[123] M. Mohammadi and S. Das, "SNN: Stacked neural networks," *arXiv:1605.08512*, May 2016.

[124] S. Irtza, V. Sethu, H. Bavattichalil, E. Ambikairajah, and H. Li, "End-To-End hierarchical language identification system," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, Mar. 2016, pp. 5820–5824.

[125] D. Faria and D. Cheriton, "Detecting identity-based attacks in wireless networks using signalprints," in *Proceedings of the ACM workshop on Wireless Security*, Los Angeles, CA, USA, Sep. 2006, pp. 43–52.

[126] Y. Sheng, K. Tan, G. Chen, D. Kotz, and A. Campbell, "Detecting 802.11 mac layer spoofing using received signal strength," in *Proceedings of the IEEE INFOCOM*, Phoenix, AZ, USA, Apr. 2008, pp. 1768–1776.

[127] Y. Zhang, Z. Li, and W. Trappe, "Evaluation of localization attacks on power-modulated challenge-response systems," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 2, pp. 259–272, May. 2008.

[128] U. Ihsan, R. Malaney, and S. Yan, "Neural network architectures for location estimation in the internet of things," in *Proceedings of the IEEE International Conference on Communications (ICC)*, Montreal, QC, Canada, Jun. 2021, pp. 1–6.

[129] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Sep. 2019.

[130] H. Fang, X. Wang, and S. Tomasin, "Machine learning for intelligent authentication in 5g and beyond wireless networks," *IEEE Wireless Communications*, vol. 26, no. 5, pp. 55–61, Oct. 2019.

[131] I. Gradshteyn and I. Ryzhik, *Table of integrals, series, and products*. San Diego, CA: Academic Press Inc., Sep. 2014.

[132] E. Lehmann and J. Romano, *Testing statistical hypotheses*, 3rd ed. New York: Springer Science & Business Media, Mar. 2006.

[133] S. Yan, B. He, X. Zhou, Y. Cong, and A. Swindlehurst, "Delay-intolerant covert communications with either fixed or random transmit power," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 129–140, Jan. 2019.

[134] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators.," *Neural networks*, vol. 2, no. 5, pp. 359–366, Mar. 1989.

[135] M. Nielson, in *Neural Networks and Deep Learning*, Determination Press, 2015.