National College of Ireland

Detection Of Replay Attacks In Autonomous Vehicles LTV Systems Using Dynamic Watermarking, Kalman Filter & Mahalanobis Distance

MSc Research Project
Cyber Security

Taher Ahmed
Student ID: 23186950

School of Computing
National College of Ireland

Supervisor: Mr. Raza Ul Mustafa

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Taher Ahmed		
Student ID:	23186950		
Programme:	MSc in Cyber Security	**Year:**	2024-25
Module:	MSc Research Project		
Supervisor:	Mr. Raza Ul Mustafa		
Submission Due Date:	12th December 2024		
Project Title:	Detection of replay attacks in Autonomous vehicle LTV systems using Dynamic Watermarking, Kalman Filter and Mahalanobis Distance		
Word Count:	8738		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Taher Ahmed

Date: 12th December 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Detection Of Replay Attacks In Autonomous Vehicle LTV Systems Using Dynamic Watermarking, Kalman Filter And Mahalanobis Distance

Taher Ahmed
23186950

Abstract

A strong replay attack in autonomous vehicle would lead to traffic disruptions, compromise of sensitive data and could also put the passenger's life at risk. Considering the evolution of autonomous vehicle and its corresponding complications of replay attack, this paper focuses on proposing a solution to detect replay attacks in autonomous vehicle's CAN bus model. Although there are various research strategies proposed on this category, the prime idea behind this research paper aims to study about the Linear Time Varying (LTV) systems which handles the core dynamics of the self driving vehicles and accountable for handling deviations in the varied conditions. This is achieved by combining dynamic watermarking with Kalman Filter and Mahalanobis Distance to estimate the state of dynamic system and handle correlated variables of CAN bus data effectively. The technique is investigated by creating a simulation bed acting as a CAN bus model to transmit messages between each nodes, and the data is extracted from Apollo Scape which consists a large scale trajectory data of urban streets with traffic flows containing vehicles, riders and pedestrians. Here, the replay attack was triggered and tested at different intervals and has resulted with higher recall rate for LTV systems with 77%, an average of 0.10 seconds detection rate, and predicting the freshness of each messages efficiently.

Keywords–Linear Time Varying (LTV), Kalman Filter, Mahalanobis Distance, Dynamic Watermarking.

1 Introduction

With a rapid development in the automobile industries, the invention of autonomous vehicles are expected to replace traditional vehicles in next few years. A fully autonomous vehicle is designed to be self aware and highly capable of making its own decision. This system is possible as the AVs function based on sensors, Control Area Network, complex algorithms, and powerful processors. Thus, a cyber attack like replay attacks pose significant threats by altering and re-transmitting messages, potentially causing severe damage to vehicle's operation. A replay attack on any of this sensitive system will arise to potential consequences such as, causing serious accidents which puts users life at risk, a corrupted CAN bus will mislead to massive traffic jams and traffic disruptions, compromise of sensitive data due to its connected systems, and this will affect the public trust and integrity towards the innovation of AV technologies. A replay attacks occurs when an attacker eavesdrops on the vehicle's communication channel, in order to intercept it and resend the fraudulent message to misuse the system. For example, when the sensor detects an obstacle in front of the vehicle and passes the communication to the actuators. Here, if the message has been captured by an attacker, then the actual message can be replayed with a fraudulent message which could lead to severe damages (Sooraj.T, 2019). Considering these factors, the research paper focuses on detecting replay attacks in autonomous vehicles by leveraging dynamic watermarking based on a Linear Time Varying (LTV) system.

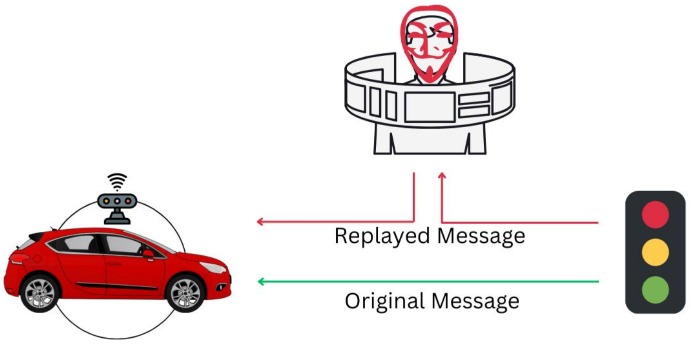

Figure 1: Replay attack in autonomous vehicle.

However, there are several research conducted on AVs for replay attacks which includes common techniques like Intrusion Detection Systems (IDS), watermarking, encryption and authentication methods. Even though these methods propose a strong detection strategies, they come with a set of limitations like ineffective handling of strong replay attacks, weak encryption with insecure time stamping approaches, failure of identifying the freshness of the message, and following Statistical Hypothesis Testing methodologies. Among these standard methodologies, dynamic watermarking has proven to be more effective in detecting replay attacks as it involves embedding secret watermarks into control signals or data streams, complicates efforts by attackers to replay data undetected. However, these previous research has explored dynamic watermarking based on Linear Time Invariant (LTI) systems, which assumes fixed dynamics and fail to address the complexities of real-world driving conditions. In LTI systems, the relationship between input and output remains linear and constant, making them suitable for simplified scenarios, such as maintaining cruise control on straight roads. However, this static nature makes LTI systems more predictable, offering attackers the potential to infer system dynamics and execute successful replay attacks (Toni.F, 2024). These studies primarily rely on Statistical Hypothesis Testing methods as a probabilistic technique for detecting replay attacks. While this method is useful for systems with known statistical models, it comes with limitations like identifying the robustness and accuracy of detecting replay attacks in a LTV system.

LTV systems have a changing behavior which can be handled by the time dependent system matrices nature of Kalman Filter. Unlike Euclidean distance, Mahalanobis distance makes the system more robust for analyzing the correlations between variables. In contrast with AI techniques, the combination of these two methods do not require excessive training data. On the other side, Heuristic approaches are less adaptable and system independent, while the Kalman Filter and Mahalanobis distance generalize to a set of conditions. The combination of Kalman Filter and Mahalanobis distance are highly efficient for state estimation and anomaly detection. Hence, this research paper focuses on enhancing effective detection of replay attacks by proposing an approach that combines Kalman Filtering, known for their ability to estimate the state of dynamic systems accurately, with Mahalanobis Distance to handle correlated variables. This will be integrated with dynamic watermarking to analyse Linear Time Varying (LTV) systems, which account for the changing dynamics of autonomous vehicles in varied conditions. The primary contributions of this research aims to:

1. Analyse the autonomous vehicle's behaviour under complicated environment conditions. Based on these dynamic factors, design and implement dynamic watermarking that is both secretive and unpredictable, to prevent attackers from replaying the data.

2. Apply Kalman Filter with Mahalanobis Distance for the LTV system to estimate the vehicle’s real-time system state and predict expected outputs. Utilize these predictions to detect and indicate replay attacks by observing discrepancies between actual and predicted outputs with minimizing delays and ensure reliable detection.

3. Implement a proof of concept within a simulated environment that models autonomous vehicle communication system, and incorporate dynamic watermarking with Kalman Filter and Mahalanobis Distance techniques to simulate replay attacks.

4. Evaluate the performance for the proposed detection mechanism with possible key metrics based on its performance and detection rate, and deliver an efficient solution to detect replay attacks in LTV systems.

2 Related Work

2.1 IDS based countermeasures

2.1.1 Intrusion detection based on bit constraint

In 2022, Kaixuan Zheng, Shihong Zou, and co-authors inestigated on a segment detection algorithm based on real-time intrusion detection of injection attacks in autonomous vehicles. This algorithm aims to improve the precision and recall rates in detecting abnormal traffic on the Controller Area Network (CAN) bus, by considering the memory limits and computational power available on multiple in-vehicle environments. This detection algorithm calculates the bit-flip rate as per each segment, and reveals the relationships between transmitted messages (Zheng et al., 2022). The approach involves various phases, such as feature analysis, model training, and real-time detection. With larger datasets, this technique resulted impressively with a combined precision and recall rate of 99.97% on detecting different types of injection attacks. However, the study identified that the algorithm's performance was poor when faced with large injection volumes, which reduced the precision and recall rates. To handle these limitations, an alternative methodology using CCID is reviewed in the following section.

2.1.2 Graph-Based IDS for CAN

In 2021, Riadul Islam, Sai Manikanta Yerram, and co-authors proposed a Graph-Based Intrusion Detection System (IDS) to detect cyber and physical attacks on the CAN bus. Their approach involves four stages of IDS layers that leverages graph-based techniques, making it reliable to detect intrusions without any changes in the CAN protocol. This system is implemented with statistical analysis to mainly detect anomalies in the communication patterns (Islam et al., 2020). The CAN messages are transformed into a graph structure, and these features are derived to segregate anomalies. This resulted with an effective detection rate of 4.76% for replay attacks, and with a less misclassification rate. Albeit the outcome being more efficient compared with other methods, this has a limitation of handling strong replay attacks which is compromised by CAN arbitration IDs. The study has investigated that the detection of replay attacks could be improved when the CAN arbitration IDs were handled separately. To manage this limitation, the next section covers an alternative technique which aims to enhance the detection of strong replay attacks in CAN bus systems.

2.1.3 CCID-CAN for autonomous vehicles

In 2019, Heng Sun, Weilin Huang, Jian Weng, and seven other researchers collaborated to design a Cross-Chain Intrusion Detection (CCID) system for the Control Area

Network (CAN) bus system in autonomous vehicles. This research addresses on a major gap in the absence of encryption and authentication mechanisms in CAN bus communications, which makes the Electronic Control Units (ECUs) vulnerable to cyberattacks (Sun et al., 2024). This framework combines the Kalman filter and Naive Bayes model to generate a Valid Bit Index (VBIN) model, which is capable of detecting anomalies within a short interval. This method has proven to be successfully detecting attacks like, spoofing, fuzzy attacks, replay attacks, and suppression attacks, with high accuracy, recall, precision, and F1-score. While the CCID-CAN system demonstrates higher precision and recall rates compared to the bit-constraint-based IDS discussed earlier, it focuses primarily on continuous message streams in time-series analysis. In this approach, each CAN ID is treated as an independent unit, which covers the limitation in detecting replay attacks by Graph based approach. Although, this article covers the limitation from other two research papers, these replay messages generated by this method are valid only for a temporary time period. Here, the attackers can exploit this by replaying legitimate data at a different point in time within the system's allowable time boundary. The IDS verifies the temporal patterns but fails to detect these delayed replayed messages, making the system vulnerable to timing-based replay attacks.

2.2 Cryptographic based detection

2.2.1 Encryption with timestamp and initialization vector

A standard authentication and encryption method was proposed by Selvamani Chandrasekaran, K.I. Ramachandran, and co-authors, with the goal of detecting and preventing replay attacks in the CAN protocol (Chandrasekaran et al., 2020). This method introduces a suppression technique by incorporating a counter or timestamp on top of encryption to ensure message authenticity. The process is categorized into authentication and encryption phases. In the authentication phase, a node authentication model is generated to distinguish between legitimate and corrupted nodes. Later, a Message Authentication Code (MAC) is enabled to ensure the integrity and authenticity of the messages. On the next step, in the encryption phase, traditional cryptographic algorithms such as Tiny Encryption Algorithm (TEA) and Secure Hash Algorithm (SHA) are used to encrypt and decrypt messages, which are incorporated with a timestamp. The results shows that the system is feasible to successfully decrypt fresh messages with valid keys and detect any replayed messages. However, this approach uses weak encryption keys with a 16-bit key length, making the system more vulnerable to brute-force attacks. Furthermore, the traditional timestamp approach creates a time window through which the attackers can exploit the system to replay messages. Considering these weaknesses, it requires more advanced encryption techniques and secure timestamping approach. An alternative approach to enhance this system is discussed in the next section.

2.2.2 Lightweight encryption & authentication for CAN

In 2023, Jie Cui, Yaning Chen, and five other researchers proposed a lightweight encryption and authentication scheme for an in-vehicle Control Area Network (CAN) bus. This approach addresses issues related to Electronic Control Units (ECUs) like consumption of time and the complexity in key distribution. The authors replaced the traditional key distribution process with Blom key management algorithm used in the CAN bus to make it more efficient (Cui et al., 2023a). The system’s effectiveness was evaluated using BAN logic and security analysis, demonstrating success in defending against attacks such as forgery and replay. While the technique provides a strong encryption framework, its reliance on the Blom key management method primarily ensures secure key distribution and communication between nodes or messages. However, the approach has failed to validate the freshness of the messages, like a message is new or replayed from old session. However, this technique might work as an additional layer of defense, but it requires enhancements like precise timestamp validation, to completely mitigate risk of replay attack.

2.2.3 A Re-Encryption scheme for autonomous vehicles

In 2023, Yun Shen, Hong Zhong, and three other researchers proposed a multilevel Electronic Control Unit (ECU) architecture with a re-encryption scheme for autonomous vehicles. This approach centers on re-encrypting all critical information, verified by the Global ECU (GECU), to prevent the transmission of malicious messages (Shen et al., 2023a). The framework also allows forwarding messages, process during single-point failures and establish cross-bus communication in a centralized gateway architecture. An enhancement made in this approach compared with previous works, is the integration of timestamps with both encryption and re-encryption phases. This works as an additional layer in preventing replay attacks, and the scheme was analyzed using a random oracle model and BAN logic to evaluate its security. This further resulted from withstanding against a variety of attacks like, sniffing, tampering, replay, collusion, flooding, and spoofing attacks. In the context of replay attacks, the Message Authentication Code (MAC) is generated during both encryption and re-encryption based on timestamps. This effectively authenticates all the nodes and prevent replayed messages from being accepted. However, while the paper highlights the novelty of its re-encryption scheme, it leaves several gaps for future research. This includes performance limitations, reduced flexibility, and the use of symmetric keys weakens the system's resistance to malicious activities. This directs to poor performance in detecting certain types of cyber attacks, a major factor addressed in next section.

2.3 Watermarking based approach

2.3.1 Dynamic Watermarking for general LTI systems

Considering an enhancement of research study based on detecting malicious sensor attacks for SISO LTI systems, a modern technique using design of dynamic watermarking for detecting malicious sensor attacks were published in 2017 by Pedro Hespanhol, Matthew Porter, and two other authors (Hespanhol et al., 2017). The previous paper was focused on systems with partial state observations and MIMO LTI systems with a full rank input matrix and full state observation. This approach is not suitable for general LTI systems which are MIMO and maintain partial state observations. This is enhanced by providing a new set of asymptotic and statistical tests in this current research study, which has proved to have the tendency to distinguish between sensor attacks and wind disturbance by developing an internal model principle framework. However, this work has been explored only for general attacks than replay attacks, which needs to be applied for other attack models. Alongside, the paper has failed to discover the robust control design in the regime of when an attack is detected.

2.3.2 Dynamic Watermarking for autonomous vehicle

In 2022, Lantian Shangguan, Kenny Chour, and four other researchers implemented a defensive technique using a dynamic watermarking scheme on a real-time autonomous vehicle sensing system. The study focused on three types of attacks, random noise, constant bias, and intermittent toggling, all of which aimed to tamper with the vehicle's yaw rate measurement at a specified linear velocity (Shangguan et al., 2022). For each attack, the system's sensitivity and responsiveness in detecting attacks were tested by using three different sizes of watermarks. The research primarily focuses on assessing how quick attacks can be identified considering the length of the watermark. The attacks were triggered through a ROS node, which changed the yaw rate measurements, reporting false data to the controller. The research resulted that the dynamic watermarking technique is effective, robust, sensitive, and responsive in detecting cyberattacks. However, a major limitation of this research is its focus on linear velocity and the use of Linear Time-Invariant (LTI) systems. The method has not been analyzed for replay attacks in Linear Time-Varying (LTV) systems, which would be critical for real-world autonomous vehicles.

2.3.3 Dynamic Watermarking using auto-correlation normalizing factor

In 2020, Matthew Porter, Sidhartha Dey, and five other researchers proposed a method for detecting replay attacks in Linear Time-Varying (LTV) systems for autonomous vehicles. This approach aims to overcome the limitations of Linear Time-Invariant (LTI) systems, which were previously used to validate measurements but struggled with time-varying effects and auto-correlation in LTV systems. The authors fixed these challenges by developing an auto-correlation normalizing factor, which is designed to eliminate auto-correlation effects from the residuals (Porter et al., 2020). This technique reinforces the statistical foundation of all the detection tests. This was followed by applying dynamic watermarking to a high-fidelity vehicle model in CarSim to further validate this approach. In their experiments, replay attacks were triggered during mid-trajectory tasks of the vehicle. The use of the LTV dynamic watermarking scheme, enhanced by the auto-correlation normalizing factor, demonstrated quick detection of these attacks, proving its effectiveness in more complex, time-varying autonomous vehicle environments.

2.3.4 Detection of replay attacks using time varying watermarking

In 2020, Matthew Porter, Pedro Hespanhol, and three other researchers proposed a Linear Time-Varying (LTV) with dynamic watermarking approach to detect generalized replay attacks in autonomous vehicles. This study introduced asymptotic guarantees along with implementable tests by designing a matrix normalization factor to handle temporary changes in the system's dynamics (Porter et al., 2020). The approach also utilized statistical tests with a sliding window method to expand the real-time attack detection limit. A vehicle equipped with the LTV dynamic watermarking scheme was used for simulation, and its performance was compared to a Linear Time-Invariant (LTI) counterpart. The results demonstrated consistent test metrics, showing that the LTV watermarking effectively detected generalized replay attacks. Notably, the method maintained a false alarm rate of no more than once per every 50 seconds of runtime.

However, despite the effectiveness of both this and the previous research methods, the approaches heavily rely on Statistical Hypothesis Testing for detecting replay attacks. While this technique works well in systems with well-understood statistical models, it has several limitations like, delays in detection, higher false positive rates, especially in noisy environments, lack of robustness under approximation errors, and failure to address key performance metrics, such as the average time to detect replay attacks.

In conclusion, the reviewed articles focus on common approaches researched to detecting replay attacks in autonomous vehicles, which are Intrusion Detection Systems (IDS), encryption and authentication techniques, and dynamic watermarking. Even though these methods propose a strong detection strategies, they come with a set of limitations such as, ineffective handling of strong replay attacks, weak encryption with insecure time stamping approaches, failure of identifying the freshness of the message, and poor detection rate. Among these standard methodologies, dynamic watermarking has proven to be more effective in detecting replay attacks as it involves embedding secret watermarks into control signals or data streams, complicates efforts by attackers to replay data undetected. The integration of dynamic watermarking into LTV systems showcase a dynamic model and active defense. This provides a realistic model for system behaviour and with this, the watermarking adds an additional layer for active probing, which makes the system resilient from a wide range of attacks. This technique makes it computationally efficient which makes the detection and mitigation of real time attacks.

While previous research has explored dynamic watermarking based on Linear Time Invariant (LTI) systems or LTV with Statistical Hypothesis Testing methodologies, therefore, this article aims to detect replay attacks in autonomous vehicle's Linear Time Varying (LTV) systems with combination of Kalman Filter, Mahalanobis distance and dynamic watermarking.

3 Research Methodology

3.1 Kalman Filter with Dynamic Watermarking

The characteristics of replay attacks follows sending previously recorded messages by altering the actual data. Here, the Kalman filter with dynamic watermarking proves efficiently in identifying the difference between the replayed data and expected input. This is a powerful mathematical algorithm significantly used in analyzing the state of a system and handles high detection rate for responses with noise and uncertainty. It starts by processing the real-time estimation including state transition model and measurement model followed by, setting up the initial values of the state vector and initialize the covariance matrices to handle the measurement noise (Liang et al., 2019). It then collects the sensor data such as speed, acceleration, orientation or position generated from the CAN bus and adds a dynamic watermark in these control signals to maintain a unique pattern.

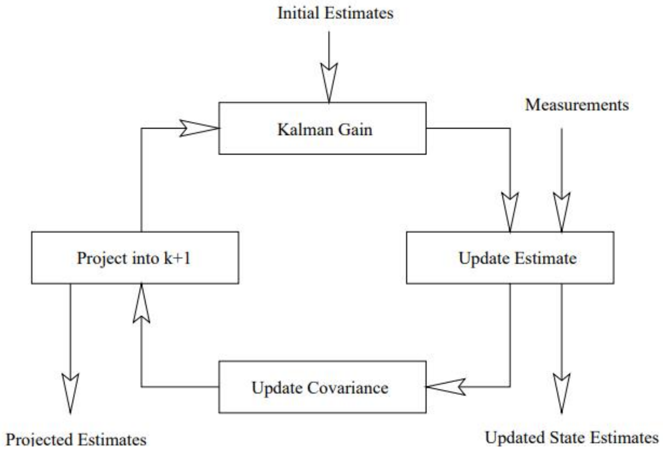

Figure 2: Kalman Filter algorithmic loop.

Kalman filter basically operates with two steps: prediction and update. In this prediction step, the next state will be predicted based on the vehicle's current state and the control inputs, and this determines the predicted state and covariance. In the update step, the received measurement will be compared with the predicted state which derives with the residuals for the prediction error. The system maintains a specific threshold and if the calculated residuals are under this level, then the normal operation will be continued. When the residuals are beyond the threshold level, it can be raised as a potential replay attack has been detected, and the iteration continues. This technique involves detecting the anomalies more efficiently and allows autonomous vehicles to enhance their security system from replay attacks (Teow.J, 2018).

In general, the sensors of the autonomous cars detect the prevailing collisions and make judgement. In order to perform this efficiently, the vehicle must predict the future metrics so that the decision could be made prior. For instance, if there are a set of cars waiting at a signal, here, the AV must detect the signal status and the distance of the last car that is in halt so that it can apply breaks at the right time. This behaviors are handled by sensors to position the objects and with the help of Kalman Filter, this uncertainty can be mitigated by predicting the future states of the vehicle. The implementation of Kalman Filter handles the errors and noise for both prediction and the measures the required tracking variables. Examining this errors, the Gaussian is characterized by the parameters mean and the width of the Gaussian, sigma squared.

$$C = 1 / \sqrt{2\Pi\sigma^2}$$

As it progresses, the following equation nu and r squared are the mean and variance for the newly observed data. This is implied to update the prior mean and variance values. This updates the prior mean with the weighted sum of the old means. This new weights are the variances from the other mean which is normalized by the sum of the weighting factors.

$$\mu = r^2\mu + \sigma^2 v / r^2 + \sigma^2$$

In the process of calculating the prediction or motion update, the vehicles are assumed to start in some position. This motion has its own uncertainty and its calculated by accumulating as the position changes. This mean value is updated by taking the previous mean and adding the motion to its movement which is u here.

$$\mu^{I} = \mu + u$$

The Kalman gain is further calculated by comparing with the observed data. This involves conjunction of Kalman with the previously estimated data and the newly observed values. This also updates process covariance based on the Kalman gain and these updates are then utilized in next circle of predictions.

3.2 Mahalanobis Distance

The data with normal distribution of mean zero and variance as one are the distance which can be calculated using Euclidean equation. On the other side, the Mahalanobis distance is an efficient statistical technique to measure the distance between data point and the distribution which is its mean value and covariance matrix. The Euclidean distance works well for a straight line separation approaches but the Mahalanobis Distance is suitable to handle the correlation between different metrics in data. This characteristics makes it robust in managing the three dimensional data distributions where the data set is clustered from a single center.

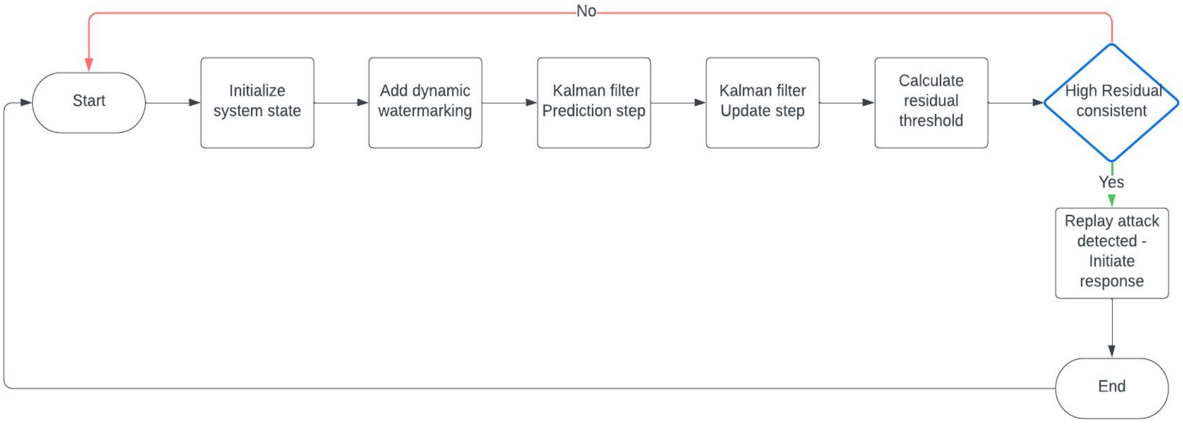
No
Start
Initialize system state
Add dynamic watermarking
Kalman filter Prediction step
Kalman filter Update step
Calculate residual threshold
High Residual consistent
Yes
Replay attack detected - Initiate response
End

Figure 3: Working model of dynamic watermarking combined with Kalman filter & Mahalanobis Distance.

In order to handle uncertain data set like the autonomous vehicle sensor data, the equation of Kalman Filter alone will not be sufficient. The data structure has multiple variances representing the position of different objects from the vehicle, such as pedestrians, signals, medians, and other vehicles. In order to predict the future state of this dynamic objects, it is highly important to combine the approach with Mahalanobis distance (Jade.D, 2024). This combined mathematical orientation handles the covariance matrix significantly by handling the complex metrics in the sensor data. The final residuals generated with Kalman Filter equations and inverted covariance matrix are used to calculate a normalized distance of Mahalanobis value to obtain the final threshold.

4 Design Specification

4.1 Control Area Network bus

This research article primarily focuses on replay attacks in autonomous vehicle's Control Area Network (CAN) bus system, as this is considered as the nervous system of AVs which acts as a communication system allowing to send and receive messages between different control units of the vehicle (Symmonds, C., 2023). This simplifies the way vehicle operates by maintaining a centralized communication system and improves reliability by avoiding the complexity of traditional wiring mechanisms. This was first developed by Bosch in 1980s by analyzing the need for reliable approach to connect the increasing number of electronic components in the system. Before the introduction of CAN bus, the vehicles were integrated with a lot of wires to connect different components making the system huge and complicated. CAN bus reduces this complexity by sharing a single communication pathway for multiple devices to communicate. This makes the system more scalable, cost effective and handles errors more effectively.

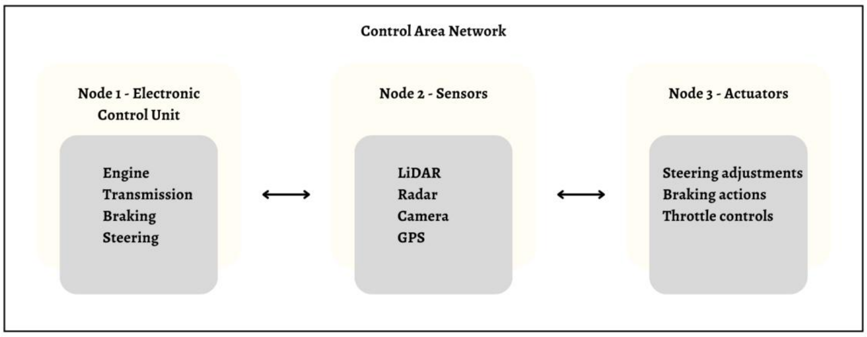

Figure 4: Control Area Network bus communication design for key components.

The core design concept of the CAN bus is to maintain a common network for different parts of the vehicle to communicate. This includes key components such as nodes, and bus. Nodes are the various components connected to the CAN bus with inter linked CAN low and CAN high wires, which handles both sending and receiving of messages. Some core nodes in CAN bus are Electronic Control Unit (ECU), sensors and Actuators as shown in figure 4. And the bus is the communication pathway which inter connects all these nodes. These components work together in sending or receiving messages and ensure the messages are transmitted properly. However, this design model is vulnerable for replay attacks as this comes with a set of limitations like the CAN bus do not a have a built in message authenticating protocols, missing timestamps, predictable data format and control signals etc., Here, marking messages with dynamic watermarking based on Kalman filter and Mahalanobis Distance to each messages makes the system feasible and prevents from replaying the old messages.

4.2 Linear Time Varying system dynamics

The LTV system in an autonomous vehicle handles core dynamics of the self driving cars which manages the dynamic changes of the vehicle with time. The key components the LTV system functions are based on sensors, actuators, Electronic Control Unit and environmental components. These primary systems integrated together plays a pivotal role in robust decision making and delivering accurate control (Li et al., 2020).

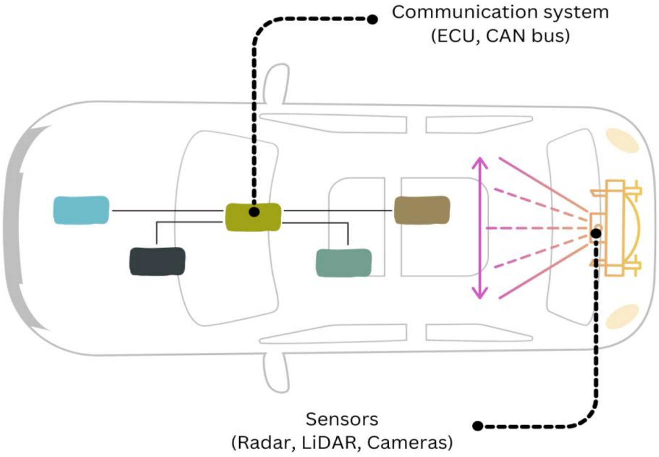

Figure 5: Core LTV system components.

Electronic Control Unit: The control unit acts as a core component that handles the communication between each nodes in a vehicle, and ensures this metrics are adjusted as per the dynamic requirements.

Sensors: Sensors in an autonomous vehicle refers to the LiDAR, cameras, GPS and radar which gathers the real-time data from the internal system and the surrounding environment.

Actuators: Actuators makes the decision based on the action received from the control system like steering, braking, acceleration to ensure the vehicle maneuvers are handled accurately aligned to the path.

Environmental component: This handles the dynamic nature of the road conditions and shares these details with control systems to manage accordingly. Based on this data, the primary controls like acceleration and brakes are adjusted.

4.3 Enhanced Architecture for CAN Bus Limitation

The CAN bus system in autonomous vehicles comes with a set of limitation which are, absence of timestamps, poor authentication and encryption, limited bandwidth, and lack of real time verification (Tara, J., 2022). In order to enhance the limitation of the CAN bus system, the architecture is improvised with integration of CAN-FD, dynamic watermarking, multiple layers with enhanced detection algorithm. The architecture is designed into layers which includes detection and processing, security layer, communication layer and gateway ECU layer. The gateway layer will monitor the abnormal traffic in the network to detect anomalies. The communication layer can be improved by adding Control Area Network Flexible data rate to enhance the network bandwidth and its efficiency. Alongside, the automotive Ethernet acts as a backbone for CAN-FD to support the higher bandwidth data like LiDAR and camera. (John, K., 2025). From the security layer is where this research is focused on by adding dynamic watermarking which is a small perturbations to detect replay attacks by comparing the threshold inconsistencies. This is handled with the help of time stamping which validates the freshness of data. The final layer is detection and processing where Kalman Filter and Mahalanobis Distance will be used to to estimate the system states, evaluate the deviation of incoming data and signal anomalies based on threshold level.

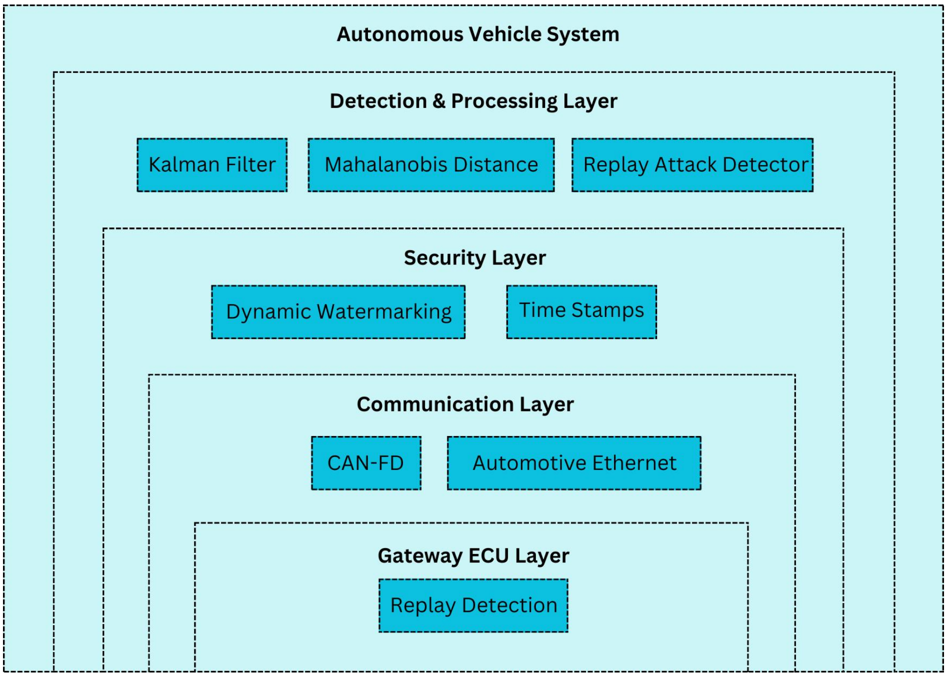

Figure 6: Enhanced CAN bus architecture to signal anomalies.

5 Implementation

5.1 Assumptions

1. This model has been designed assuming the messages are transmitted between each nodes in CAN bus model which are, Sensor, ECU and Actuator.
2. The data set used provides sensor data which is assumed to be transmitted between the sensor nodes to actuator nodes. Here, the actuator nodes collects the messages and performs the action.
3. The simulation environment is design based on assuming the CAN bus model handling only the sensor data which circles information about trajectory details such as people, vehicles, obstacles etc., around the vehicle.

5.2 Set up simulation environment

5.2.1 Prepare CAN bus nodes using Python can

Considering the key components of LTV system in an autonomous vehicles, three nodes, ECU, Actuators and Sensors within a CAN bus are created to send and receive messages between each units. This simulation environment set up is generated using Python and its dedicated library 'python-can' to depict CAN bus model (Python Can, 2023). Here, while the nodes send and receive messages and when the message belong to that particular node, it then accepts and delivers the task. For example, if the vehicle identifies a curve ahead of it, it transmits the message through sensor to both ECU and actuator. The actuator approves this request and handles the maneuver to avoid any damage.

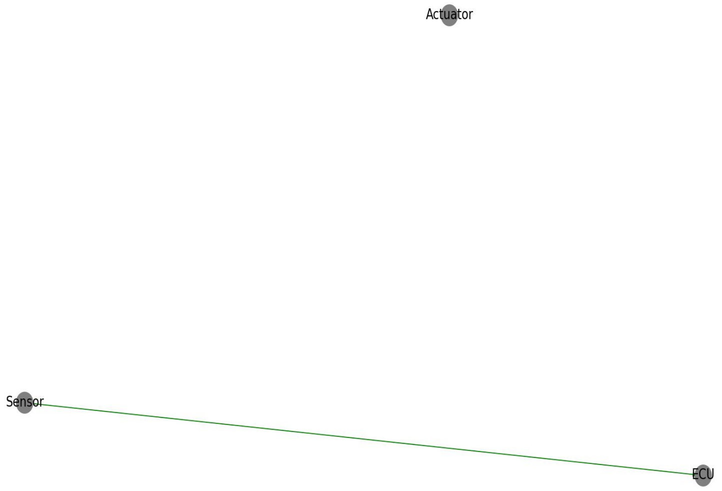

Figure 7: Message transfer between each nodes in CAN bus.

5.2.2 Handle sensor dataset for message transfer

In order to simulate message transfer between each nodes, a dataset from A*pollo Scape* has been integrated (Apollo Scape, 2023). This dataset provides a large scale trajectory dataset of urban streets consisting of camera based images, LiDAR scanned point clouds and manually annotated trajectories along with traffic flows containing vehicles, riders and pedestrians. This overall sensor data of more than 400,000 data points are used as message transfer between sensor nodes to ECU or Actuator nodes in the simulation environment to handle the dynamics of the vehicle maneuver based on detected objects around the vehicle to avoid any collision.

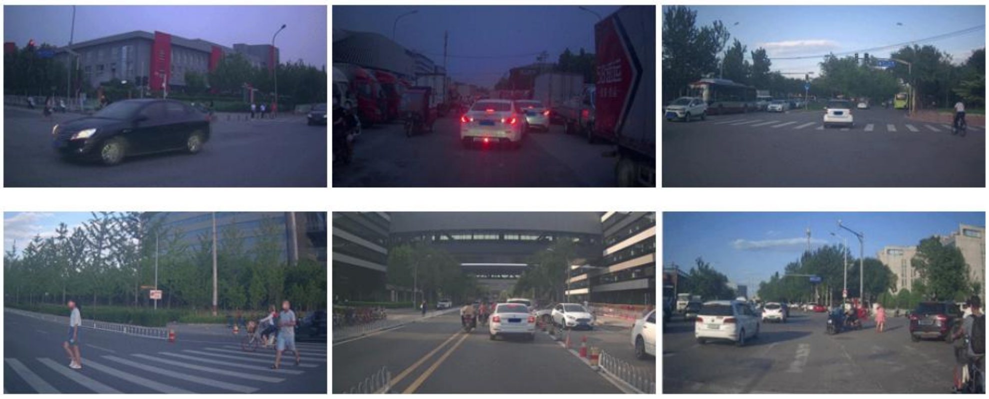

Figure 8: Camera based images of trajectory sensor from the apploscape dataset.

The dataset provides overall details of five different object types such as, small vehicles, big vehicles, pedestrian, motorcyclist and bicyclist, and other variants (ApolloScapeAuto, 2018). This data structure is categorized with separate columns as, frame id, object id, object type, position x, position y, position z, object length, object width, object height and heading. Considering the amount of dataset, the first step of pre-processing involved inspecting the raw data to ensure data with proper delimiter, removing null values or inappropriate values from the dataset and converting the data type into float to pass into CAN bus messages. This ensures that the dataset has valid and appropriate data to maintain the reliability and accuracy for simulation of messages.

Furthermore, in order to transmit the data through CAN messages, the range value exceeding 0 to 255 bytes were encoded to split each values and transmit across multiple bytes. This technique is known as Byte Splitting which allows the larger values into sequence of bytes and later be decoded to reassemble to fetch the actual data. Hence, each rows are converted into floating values and transformed into multiple bytes to make sure the bytes fit within the 0 to 255 bytes range and ensure not breaking the actual value.

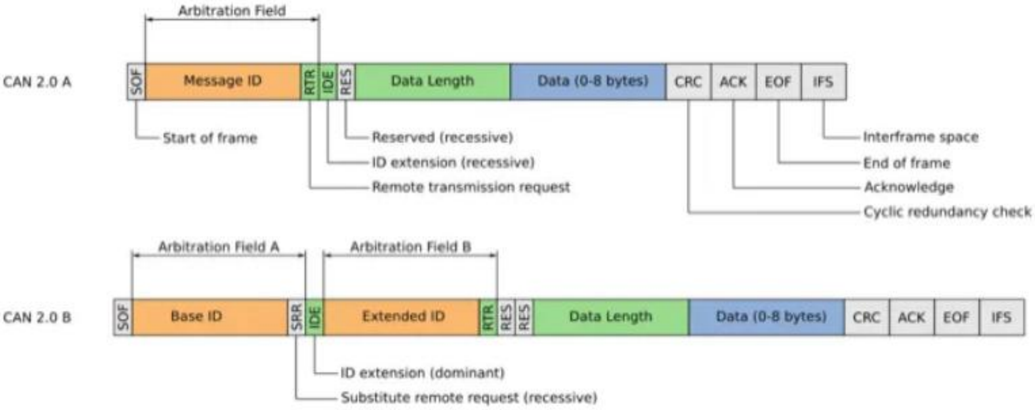

Figure 9: CAN message byte splitting.

For instance, both the messages in the figure 8 shows a maximum payload of 8 bytes. The standard CAN data rates in vehicles widely range between 250 kb/s to 1 MB/s. However, the heavy equipment AVs usually follow 29-bit identifiers whereas the other vehicles allow 11-bit identifier to manage their custom defined messages (Horton.M, 2019). However, to maintain the convenience and reliability, the CAN bus follows byte splitting to maintain the payload to be within 8 bytes and thus this characteristics is handled by default in python bus library. Hence, all the messages are converted to 8 byte in order to maintain the message communication from sensor to other nodes in the CAN bus simulation.

5.3 Perform and detect replay attack

In order to simulate a replay attack within the message transmission between sensor and actuator nodes, the previous sensor messages sent into CAN bus are captured and these messages are replayed with a varying watermark at different time intervals. This helps in exploiting the predictable nature of the messages transmitted between each components. The dynamic watermarking applied here makes the system complex by setting varied intervals when messages are re-transmitted (Rushikanjaria, 2021). This simulation scenario depicts the attackers nature on trying to replicate the legitimate traffic patterns to exploit the system. The actuators collecting these messages will respond inappropriately in failure of detecting or alerting a replay attack, which leads to several critical safety issues like brake failure or unintentional acceleration.

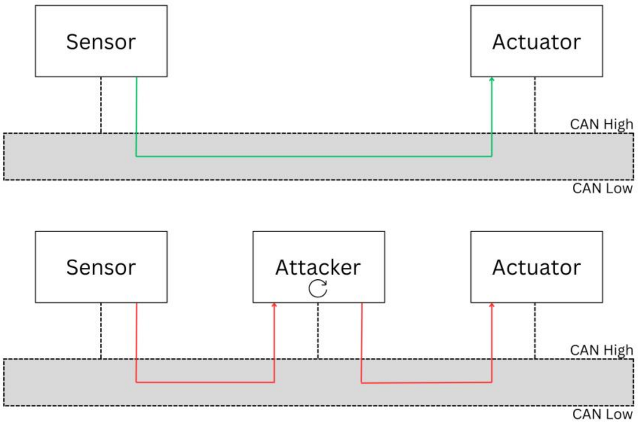

Figure 10: Simulated replay attack concept model.

5.3.1 Kalman Filter

The attack is detected using Kalman filter where various sensor state variables based on positions are set to reasonable values by constructing the object to specify the size of the state vector with dimension X and the size of the measurement vector which is dimension Z, which will be used to check the sign values for various metrics. This mathematical model analysis the sensor state of the dynamic system and update this predictions based on the results obtained. In the two primary steps of the Kalman filter, the prediction step predicts the next covariance sensor data by estimating the current sensor data and the process model. The update state updates this estimation and covariance as per the measurement by comparing the prediction of the original measurement (Lacey.T, 2022). Here, the measurements are derived from the objects such as vehicle or pedestrians and its corresponding position_x, position_y and position_z in order to estimate the next sensor input.

Assuming the positions of the objects, the covariance matrix at a time interval of *k* can be written as,

$$P_k = E\,[e_k\ e^{T}{}_k] = E\,[(x_k - x^{\wedge}{}_k)\,(x_k - x^{\wedge}{}_k)^T]$$

Based on this estimation and the Kalman gain in the sensor inputs, the update equation for the new estimation will be,

$$x^{\wedge}{}_k = x^{\wedge I}{}_k + K_k\,(z_k - H\,x^{\wedge I}{}_k)$$

Based on the above principles, the Kalman filter equation is derived by associating the measurement prediction covariance and thus,

$$K_k = P^I{}_k\,H^T\,(H\,P^I{}_k\,H^T + R)^{-1}$$

Based on this mathematical approach, the Kalman filter using Python is derived with ‘filter py’ (Python Filterpy, 2023) which handles the algorithm by initializing key elements which are, State vector (x) in order to define the current state of the sensor data, State transition matrix (F) to determine how the vehicle evolves over time, Process noise (Q) to analyze the uncertainty in the model, Measurement matrix to map the space of the measurement with the state vector, Measurement noise (R) to detect the uncertainty in the sensor data and covariance matrix (P) to record this uncertainty.

5.3.2 Mahalanobis Distance

However, in order to handle complex dynamics like trajectory sensor data and calculating its measurements based on position_x, position_y and position_z, the Kalman filter might result with less detection rate and higher false positive rates due to its prediction nature. Thus, the Mahalanobis distance calculates the distance between two measurements in a multivariate space. For correlated variables like the sensor data used, where the measurements have more than three variables, becomes complicated to plot in a regular 3D space (Stephanie, 2017). Here, the implementation of Mahalanobis distance resolves this measurement complications by accurately measuring the distance between two points. This distance is measured in basis of centroid, which measures the central tendency in a multi variant space where means from all the variables are intersected. The mathematical algorithm of the Mahalanobis distance is defined as,

$$d_{Mahalanobis} = [(x^B - x^A)T * C\ -1 * (x^B - x^A)]0.5$$

Here, the x^A and x^B are the difference between the objects, C defines the covariance matrix and T flips the matrix over its diagonal. The residuals generated from the Kalman filter and the inverted covariance matrix are derived to obtain a Mahalanobis distance which is further compared with a certain threshold level in order to detect a potential replay attack. This threshold level is set based on analyzing the obtained results and the characteristics of the trajectory sensor data.

6 Evaluation

The reason behind using the approach of dynamic watermarking with Kalman filter combined with Mahalanobis distance in detection of replay attack in autonomous vehicle LTV system is to predict robust system states and analyze precise deviation of the upcoming measurements. This technique has a higher efficiency to handle LTV system dynamics and detects deviation in the generated residuals to detect potential replay attacks in a short span of time. The major limitations observed from previous research techniques includes, poor performance, delay in detection rate, and failure of identifying freshness of message. Alongside, most of these metrics has been evaluated for LTI systems of autonomous vehicles but not considered for an advanced dynamic nature of AV's system states. Considering these limitations along with other key metrics based on dynamic nature of the system are evaluated and discussed below.

6.1 Time taken to detect replay attack

The detection rate is derived based on the ratio of number of anomalies detected with the total number of anomalies present (John.J, 2023). The approach has resulted with significant results in detecting a replay attack between the range of 0.00 to 0.20 seconds which proves the technique detects a potential anomaly in a short span of time. This prompt detection rate is highly important in real time scenarios to ensure the safety of the vehicle and the overall performance of the system. This combination of Kalman filter with Mahalanobis distance has overtaken the detection rates of other traditional approaches when it comes to how quick an attack could be identified by the system. Comparing this detection rate with other traditional methods studied in research paper (Sun et al., 2024) such as VBIN, KF-NB and CCID-CAN, the detection rate produced by this approach has resulted with an efficient rate. The average detection rate of VBIN is 0.10, KF-NB is 0.49 and CCID-CAN is 0.60, whereas the detection time using dynamic watermarking with Kalman filter and Mahalanobis distance results average of 0.10 seconds.

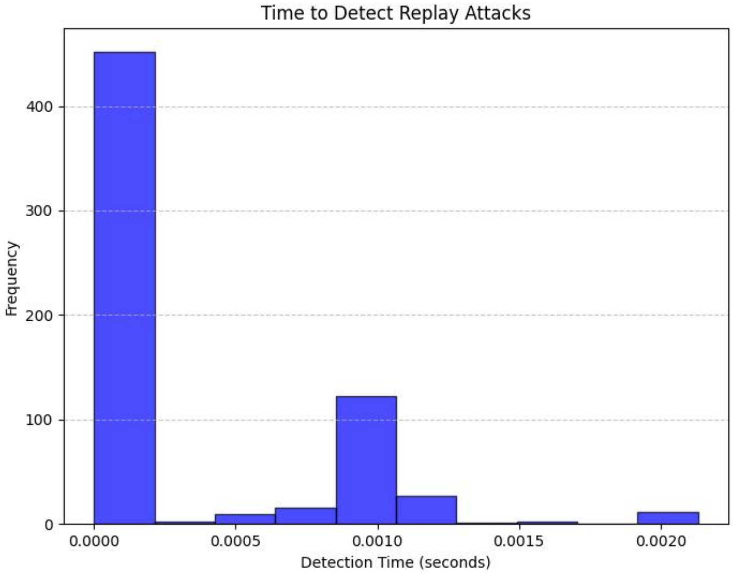

Figure 11: Replay attack detection rate.

6.2 Overall Performance

As shown in fig. 11, the technique has resulted with 42% accuracy, 34% precision, 77% recall rate, and 47% F1-Score. From the overall performance, it has resulted with higher true positive rate of 77%, which significantly improves handling multidimensional states of an AV (Jwo et al., 2023).

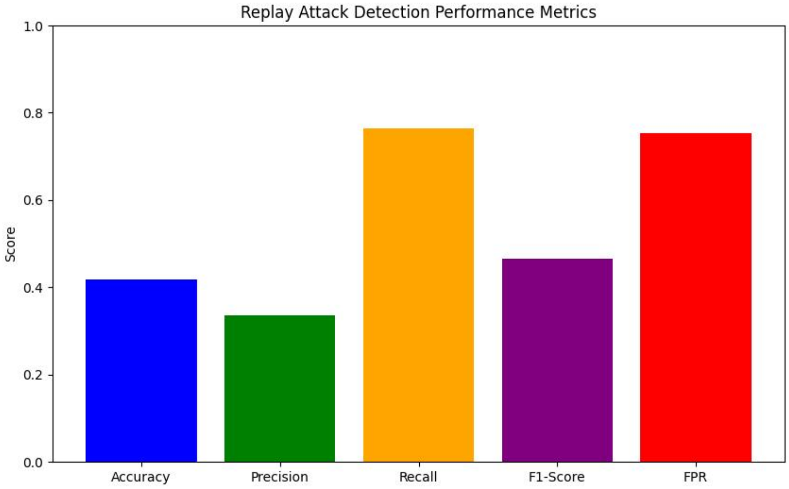

Figure 12: Performance of detecting replay attack.

This overall performance proves to be a lightweight approach for embedded systems and this combination of Kalman filter and Mahalanobis distance proves an interpretable approach which showcases adaptability real time AV systems. However, this approach has resulted with fall in precision which produces higher false positive rates and the accuracy with 42% proposes an imbalance in true and false positive resulting with a sub-optimal overall performance. Albeit its downfall in few metrics, the performance has resulted with higher recall rate and computational efficiency which makes this approach handle effectively in autonomous vehicle's dynamic environment and its correlation with system state factors.

6.3 Mean Squared Error (MSE) of Residuals

The MSE of residuals is evaluated to differentiate between the Kalman Filter predictions from the observed values. The metric has resulted with high efficiency for multi dimensional residuals and with combination of Mahalanobis distance has obtained robustness for identifying the malicious patterns accurately (Elshaer et al., 2021).

$$\mathrm{MSE} = \Sigma(\mathrm{yi} - \mathrm{pi})2\mathrm{n}$$

The flat line in the graph indicates the stabilization of Kalman Filter in order to maintain the accurate predictions. The spikes over time indicates the replayed messages as an abnormal patterns to measure the track deviation and the steady increase and fall projects the identification of replay attacks and the false positives rates. Its overall efficiency has resulted with identifying the freshness of each messages with high accuracy and not overlapping most of the replayed messages.

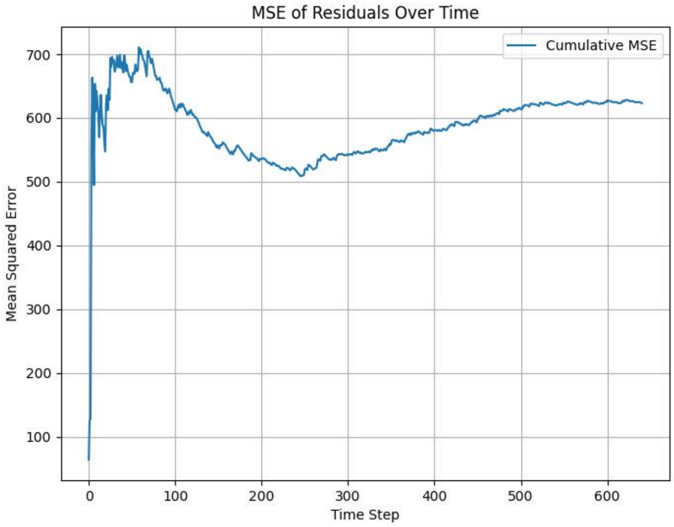

Figure 13: MSE of Kalman Filter Residuals.

6.4 Normalized Mahalanobis Distance Distribution

The normalized Mahalanobis distance evaluates the deviation of the residuals from mean distribution and scales this value to a standard threshold in order to detect deviations. This is derived based on calculating the mean distance and standard distance as derived from the equation below.

$$\mathrm{NMD} = \mathrm{MD} / \sqrt{\ } \mathrm{D}$$

As shown in fig. 13, most of the distances are normalized and falls under the threshold level which proves the Kalman filter has significantly reduced residual errors in normal operations. Alongside, the spikes close to threshold level indicates anomalies and has effectively handled in noisy environment. However, fine tuning the threshold level based on each dataset could result with more accurate results.

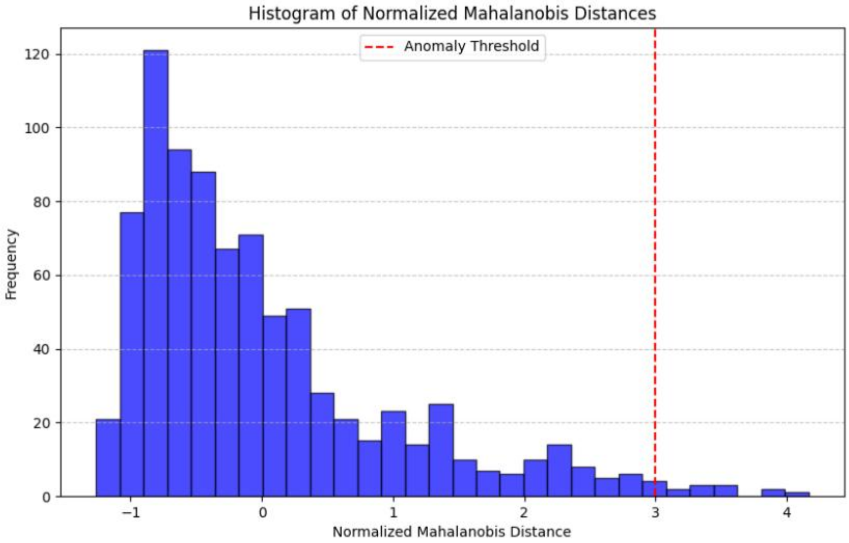

Figure 14: Normlaized Mahalanobis Distance.

6.5 Detection Score

The Mahalanobis distance is calculated to handle correlated variables in the sensor data set which operates with two measurements in a multivariate space. This is derived by combining with Kalman filter which enforces a robust framework in detecting replay attacks for complex dynamics of the LTV systems. The obtained results demonstrates handling the fluctuations of the data efficiently and this provides a highly scalable environment to work with larger datasets. However, the approach has resulted with inefficiency in clustering which makes it difficult to identify the patterns between normal signals and replayed messages. This can be further optimized by handling the noise frequency to enhance the stability of data by maintaining a lesser false positive rates.

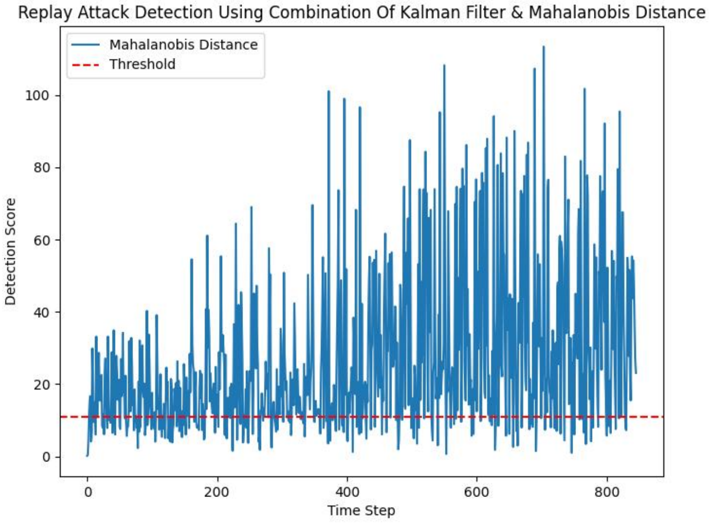

Figure 15: Mahalanobis Distance Detection Score.

6.6 False Positive Rate in Noisy Environment

Considering the noise factor from the obtained data, the false positive rate has resulted with above 70%. Albeit resulting with a higher false positive value, based on the detection speed evaluated, it determines that 77% of the attacks are detected which captures most of the genuine attacks effectively. Alongside, for an application like autonomous vehicle dealing with safety measure, the risk of failing to detect a potential attack is severe than risk of predicting a false positive rate, which focuses on sensitivity of the attack more than its specificity. However, the higher false positive rate affects the system's reliability and effectiveness, and this can be improved by integrating dynamic threshold detection into the existing technique and test the approach with various data set models.

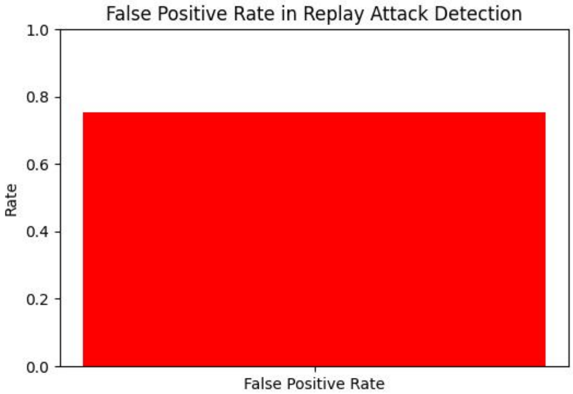

Figure 16: False Positive Rate in Noisy Environment.

In future studies, state of art approaches like Recurrent Neural Networks (RNN) models, Hidden Markov Models (HMM), Cryptographic authentication methods, and Hybrid models can be used to benchmark the False Positive Rate performance. Through these advanced methodologies, different test scenarios can be collected and compare the results for all these state of art approaches. By evaluating these metrics in a large scale for each implementation will result in analyzing the best detection rate. This research paper has focused only on combination of dynamic watermarking with Kalman Filter and Mahalanobis Distance, whereas the suggested alternative approaches collectively will result with better performance analysis and, help to analyze the trade-offs based on different applications.

6.7 Discussions

The detection of replay attack for Linear Time Varying systems in autonomous vehicles has resulted significantly in detection time with an average of 0.10 seconds and with detection rate of 77% which is also the true positive rates from the overall sensor data. Considering the dynamic metrics and noisy environment, the designed framework has proved with better performance, efficient detection rate, and capability in identifying the freshness of message through resulted MSE of residuals and normalized MD.

However, due to the clustered data and noise factor, the resulting false positive rate has been comparatively high. This limitation from this methodology can be improvised by first identifying the patterns in the false positive rates through multiple channels like system logs and event handlers. Ensure high quality of data is used to generate test scenarios and make the Kalman Filter dynamic to improve the accuracy of estimating the state of vehicle, and fine tune the sensitivity in the deviations of Mahalanobis Distance. This technique can be optimized by maintaining a dynamic threshold to handle the multidimensional states effectively and test on various other metrics to enhance this solution. Furthermore, multiple detection layers can be implemented like Ensemble methods in order to minimize the false positive alarms. Alongside, a hybrid based detection model can be implemented using various state of art techniques as discussed in above section 6.6 to achieve high detection accuracy, less false positive rates, and improve the robustness of detection mechanism.

On the other side, enhancing the CAN bus architecture considering the control systems and cybersecurity components relies on the trade-offs between system complexity and performance improvements. The architecture requires higher computation compared with current system to improvise the performance which leads to time delay and higher cost in addition of resources. An advanced system needs to be equipped to maintain the dynamics of the environment which will result in huge chunk of code base and maintainability. When it comes to scalability of the application, an optimized performance to maintain load balancing is efficient which requires intricate distribution system with advanced hardware and networking. However, these trade-offs are achievable by maintaining a modular design concept to handle the complexity without affecting the performance and, combine lightweight models with advanced algorithms to reduce the complexity.

7 Conclusion and Future Work

The core idea of this research paper was to investigate on a framework to detect replay attack in autonomous vehicle's LTV system using Kalman Filter and Mahalanobis distance integrated with dynamic watermarking. The research has demonstrated promising results in identifying the replayed messages for the vehicle's dynamic nature with 77% of recall rate to detect a potential attack, an average of 0.10 seconds detection rate, and predicting the freshness of each messages efficiently. This could possibly provide an overview for self driving vehicles to manage their CAN bus model based on LTV systems in identifying anomalies at a quick rate and higher precision for various non linear metrics. However, this approach has resulted with higher false positive rates due to the noisy environment in sensor data. This framework can be further enhanced by evaluating the thresholds dynamically based on the current and previous states of the vehicle. Alongside, the paper focuses only on a certain type of sensor data set and thus, various other data sets can be evaluated and tested to propose a more optimal solution. With further fine tuning using dynamic threshold, this technique will reduce the false positive rates to handle the multidimensional states effectively and maintain a rigid balance between the sensitivity and specificity of the CAN bus data in terms of detecting the replay attack.

References

Apollo Scape (2023). Apollo Scape. [online] apolloscape.auto. Available at: https://apolloscape.auto/trajectory.html.

ApolloScapeAuto (2018). dataset-api/trajectory_prediction at master · ApolloScapeAuto/dataset-api. [online] GitHub. Available at: https://github.com/ApolloScapeAuto/dataset-api/tree/master/trajectory_prediction.

Symmonds, C. (2023). CAN Bus Protocol: The Ultimate Guide (2022) | AutoPi. [online] AutoPi.io. Available at: https://www.autopi.io/blog/can-bus-explained/.

Cansiz, S. (2023). Mahalanobis Distance & Multivariate Outlier Detection in R | Built In. [online] builtin.com. Available at: https://builtin.com/data-science/mahalanobis-distance.

Chandrasekaran, S., Ramachandran, K.I. and Adarsh, S. (2020). Avoidance of Replay attack in CAN protocol using Authenticated Encryption. [online] ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/9225529.

Cui, J., Chen, Y., Zhong, H., He, D., Wei, L., Bolodurina, I. and Liu, L. (2023a). Lightweight Encryption and Authentication for Controller Area Network of Autonomous Vehicles. IEEE Transactions on Vehicular Technology, pp.1–15. doi: https://doi.org/10.1109/tvt.2023.3281276.

Cui, J., Shen, Y., Zhong, H., Zhang, J. and Liu, L. (2023b). A Multilevel Electronic Control Unit Re-Encryption Scheme for Autonomous Vehicles. IEEE Transactions on Intelligent Transportation Systems, [online] 25(1), pp.104–119. doi: https://doi.org/10.1109/tits.2023.3309817.

Toni, F. (2024). Autonomous Cars & Cyber Risks. [online] HTTPCS Blog. Available at: https://blog.httpcs.com/en/autonomous-cars-cyber-risks/.

Jade, D. (2024). Demystifying Mahalanobis Distance: The Secret Weapon for Data Outliers. [online] Medium. Available at: https://medium.com/@TheDataScience-ProF/demystifying-mahalanobis-distance-the-secret-weapon-for-data-outliers-9cc58e87cdf8 [Accessed 11 Dec. 2024].

Elshaer, A.M., Elrakaiby, M.M. and Harb, M.E. (2018). Autonomous Car Implementation Based on CAN Bus Protocol for IoT Applications. [online] IEEE Xplore. doi: https://doi.org/10.1109/ICCES.2018.8639206.

Frost, J. (2021). Mean Squared Error (MSE). [online] Statistics By Jim. Available at: https://statisticsbyjim.com/regression/mean-squared-error-mse/.

Hespanhol, P., Porter, M., Vasudevan, R. and Aswani, A. (2017). Dynamic watermarking for general LTI systems. arXiv (Cornell University), [online] pp.1834–1839. doi: https://doi.org/10.1109/cdc.2017.8263914.

Horton, M. (2019). What can a CAN bus IMU do to make an autonomous vehicle safer? [online] Medium. Available at: https://medium.com/@mikehorton/what-can-a-can-bus-imu-do-to-make-an-autonomous-vehicle-safer-e93f748569f6 [Accessed 11 Dec. 2024].

Islam, R., Refat, R.U.D., Yerram, S.M. and Malik, H. (2020). Graph-Based Intrusion Detection System for Controller Area Networks. IEEE Transactions on Intelligent Transportation Systems, pp.1–10. doi: https://doi.org/10.1109/tits.2020.3025685.

John, J. (2023). Detection Rate - an overview | ScienceDirect Topics. [online] www.sciencedirect.com. Available at: https://www.sciencedirect.com/topics/computer-science/detection-rate.

Jwo, D.-J. and Biswal, A. (2023). Implementation and Performance Analysis of Kalman Filters with Consistency Validation. Mathematics, [online] 11(3), p.521. doi: https://doi.org/10.3390/math11030521.

Kaiser, A., Schenck, W. and Möller, R. (2022). Distance Functions for Local PCA Methods. [online] Available at: https://www.ti.uni-bielefeld.de/downloads/publications/kaiser_esann10.pdf [Accessed 11 Dec. 2024].

Sooraj, T. (2019). What Is a Replay Attack? [online] Kaspersky.com. Available at: https://www.kaspersky.com/resource-center/definitions/replay-attack.

Lacey, T. (2022). Tutorial: The Kalman Filter. [online] Available at: https://web.mit.edu/kirtley/kirtley/binlustuff/literature/control/Kalman%20filter.pdf.

Lantian Shangguan, Chour, K., Woo Hyun Ko, Kim, J.-W., Gopal Krishna Kamath, Bharadwaj Satchidanandan, Swaminathan Gopalswamy and Kumar, P.R. (2023). Dynamic Watermarking for Cybersecurity of Autonomous Vehicles. IEEE Transactions on Industrial Electronics, 70(11), pp.11735–11743. doi: https://doi.org/10.1109/tie.2022.3229333.

Li, Y., Chen, X. and Mårtensson, J. (2020). Linear Time-Varying Model Predictive Control for Automated Vehicles: Feasibility and Stability under Emergency Lane Change. IFAC-PapersOnLine, 53(2), pp.15719–15724. doi: https://doi.org/10.1016/j.ifacol.2020.12.052.

Liang, C., Wen, F. and Wang, Z. (2019). Trust-based distributed Kalman filtering for target tracking under malicious cyber attacks. Information Fusion, 46, pp.44–50. doi: https://doi.org/10.1016/j.inffus.2018.04.002.

Porter, M., Dey, S., Joshi, A., Hespanhol, P., Aswani, A., Johnson-Roberson, M. and Vasudevan, R. (2020a). Detecting Deception Attacks on Autonomous Vehicles via Linear Time-Varying Dynamic Watermarking. 2020 IEEE Conference on Control Technology and Applications (CCTA). doi: https://doi.org/10.1109/ccta41146.2020.9206278.

Porter, M., Hespanhol, P., Aswani, A., Johnson-Roberson, M. and Vasudevan, R. (2020b). Detecting Generalized Replay Attacks via Time-Varying Dynamic Watermarking. IEEE Transactions on Automatic Control, pp.1–1. doi: https://doi.org/10.1109/tac.2020.3022756.

Python Can (2023). python-can 4.3.1 documentation. [online] python-can.readthedocs.io. Available at: https://python-can.readthedocs.io/en/stable/.

Python Filterpy (2023). KalmanFilter — FilterPy 1.4.4 documentation. [online] filterpy.readthedocs.io. Available at: https://filterpy.readthedocs.io/en/latest/kalman/KalmanFilter.html.

Rushikanjaria (2021). Designing a Self-Driving car simulation using python. [online] Analytics Vidhya. Available at: https://medium.com/analytics-vidhya/designing-a-self-driving-car-simulation-using-python-38dcac8136c6.

Stephanie (2017). Mahalanobis Distance: Simple Definition, Examples. [online] Statistics How To. Available at: https://www.statisticshowto.com/mahalanobis-distance/.

Sun, H., Huang, W., Weng, J., Liu, Z., Tan, W., He, Z., Chen, M., Wu, B., Li, L. and Peng, X. (2024). CCID-CAN: Cross-Chain Intrusion Detection on CAN Bus for Autonomous Vehicles. IEEE Internet of Things Journal, [online] 11(15), pp.26146–26159. doi: https://doi.org/10.1109/jiot.2024.3393122.

Synopsys (2022). What is an Autonomous Car? – How Self-Driving Cars Work | Synopsys. [online] www.synopsys.com. Available at: https://www.synopsys.com/glossary/what-is-autonomous-car.html.

Teow, J. (2018). Understanding Kalman Filters with Python. [online] Medium. Available at: https://medium.com/@jaems33/understanding-kalman-filters-with-python-2310e87b8f48.

Wang, P., Huang, X., Cheng, X., Zhou, D., Geng, Q. and Yang, R. (2019). The ApolloScape Open Dataset for Autonomous Driving and its Application. IEEE Transactions on Pattern Analysis and Machine Intelligence, pp.1–1. doi: https://doi.org/10.1109/tpami.2019.2926463.

Zheng, K., Zou, S., Xu, G. and Bi, Z. (2022). Segment Detection Algorithm: CAN bus intrusion detection based on Bit Constraint. Segment Detection Algorithm: CAN bus intrusion detection based on Bit Constraint, [online] pp.450–456. doi: https://doi.org/10.1109/wowmom54355.2022.00070.

John, K. (n.d.). CAN FD Explained - A Simple Intro [2022 | The #1 Tutorial]. [online] Available at: https://www.csselectronics.com/pages/can-fd-flexible-data-rate-intro.

Tara, J. (2025). CAN Network Protocol: Pros, Cons, Applications. [online] Available at: https://www.gridconnect.com/blogs/news/can-network-protocol-advantages-disadvantages-application-examples?srsltid=AfmBOoqi6RP0zrpq1CZpirVt1fAJU4kXqWP80lhIimRuPwwiurZ5r-um