


Implementation of EWMA Algorithm in the Analysis of Security Attacks

*¹Şükrü OKUL, ²Fatih KELEŞ, ³Muhammed Ali AYDIN

¹ Corresponding Author, TÜBİTAK-BİLGEM, Türkiye, sukruokul@tubitak.gov.tr 

² Department of Computer Engineering, Istanbul University - Cerrahpasa, Türkiye, fkeles@iuc.edu.tr 

³ Department of Computer Engineering, Istanbul University - Cerrahpasa, Türkiye, aydinali@iuc.edu.tr 

Abstract

This study analyzes the detection of security attacks on smart vehicles using the Exponentially Weighted Moving Average (EWMA) algorithm. We employed synthetically generated datasets, consisting of 80% non-attack and 20% attack scenarios. Various smoothing parameters (α) were tested within the EWMA framework, specifically at values of 0.8, 0.7, and 0.6, with 0.7 yielding the most promising results. In our analysis, we normalized the selection function in the EWMA algorithm based on expert evaluations to establish the impact of different factors on anomaly detection. Specifically, we assigned weights of 24% to RPM, 40% to speed, and 18% each to fuel quantity and accelerator pedal position. The results demonstrate that the EWMA algorithm can effectively issue warnings for vehicles under potential attack, enabling proactive measures to mitigate security risks. This research contributes to enhancing the safety and reliability of smart vehicles by facilitating timely responses to detected security threats.

Keywords: Security Attacks; Smart Vehicles; EWMA

1. INTRODUCTION

Security attacks pose a significant threat, particularly to smart devices, which are increasingly interconnected through the Internet of Things (IoT). As the importance of security in modern applications grows, ensuring the safety of smart vehicles has become paramount. This study examines smart vehicles, focusing on their communication protocols, infrastructure, and the various attack scenarios, causes, and consequences they may encounter.

The literature on smart vehicle security highlights critical areas such as Vehicle-to-Vehicle (V2V) communication frameworks, the challenges inherent in V2V data transmission, and the cybersecurity vulnerabilities present in these systems [1-2]. Research indicates that smart vehicles can be attacked both directly and indirectly through their Control Area Network (CAN) and via radio frequencies [3]. A comprehensive understanding of these attack types, as well as the classification of cyber threats, is essential for developing effective security measures.

In this context, the Exponentially Weighted Moving Average (EWMA) algorithm has been identified as a promising tool for detecting anomalies in vehicle data. Although the EWMA algorithm has historical significance, emerging studies suggest its growing analytical value in cybersecurity

applications [4-5]. This algorithm is particularly effective in identifying subtle changes in data, making it suitable for detecting security attacks in smart vehicle systems [6].

To validate the applicability of the EWMA algorithm in smart vehicle security, experiments were conducted using two synthetically generated datasets: one comprising 80% non-attack data and 20% attack data, and the other consisting of 70% non-attack and 30% attack data. The simulated datasets represented either normal operating conditions or data subjected to various attack types, as detailed in the literature [7-8].

The findings reveal that the EWMA algorithm can successfully identify security attacks by monitoring the CAN network of smart vehicles. The algorithm achieved a high success rate in detecting anomalies, but further improvements could be realized by optimizing key parameters such as rpm, speed, throttle, and fuel consumption, as well as exploring alternative algorithms for comparison.

2. LITERATURE REVIEW

Research on vehicle-to-vehicle (V2V) communication has gained significant traction in recent years, particularly in the context of transmitting vehicle information effectively [9-

10]. Methods for disseminating this information generally fall into two categories: centralized and decentralized systems. Centralized systems involve infrastructure-to-vehicle (I2V) communication or mobile communications, where vehicles collect and relay information through roadside units or mobile terminals, respectively [11].

Conversely, decentralized V2V communication allows for direct information exchange between vehicles, which is particularly beneficial in emergency situations, eliminating the need for additional infrastructure like roadside units and base stations [12]. Traditional centralized systems often impose significant burdens on communication infrastructure and data centers, prompting a shift toward decentralized models. Vehicle identification data obtained through these mechanisms can be leveraged to support Driving Safety Support Systems, potentially alleviating congestion and enhancing road safety. However, existing identification distribution systems face challenges such as limited-service areas, low delivery efficiency, and delays in data transmission [13].

In high-traffic environments, V2V communication facilitates efficient information exchange among vehicles. For instance, when the number of vehicles falls within a specified range, V2V communication is employed using Geocast techniques. Geocast refers to location-based data transmission in an ad-hoc network, replacing traditional node IDs with geographic information [14]. Key factors influencing communication between nodes in such networks include the target node's location and the selected transmission path.

I2V communication is particularly useful in densely populated areas or at traffic intersections, where roadside units are deployed. When a vehicle is within a predetermined distance from a roadside unit, I2V communication is activated. The roadside unit's location information is derived from digital maps, allowing the ego vehicle to assess its position relative to the roadside unit to determine the appropriate communication mode [15].

Mobile communication serves in scenarios with low vehicle density or where direct V2V communication may lead to network congestion. In cases where the number of nearby vehicles is below a specified threshold or exceeds a certain limit, mobile communication becomes the preferred option. This mode is particularly suitable for delivering non-urgent information and operates over 3G networks, enabling extensive coverage and the flexibility of pull-type communication based on driver needs [16].

3. MATERIALS AND METHODS

The Exponentially Weighted Moving Average (EWMA) algorithm was first introduced under the name Geometric Moving Average (GMA) [17]. Initially, it saw limited application outside a few studies [18-19]. However, its analytical significance began to grow in the latter half of the 1980s, leading to its adoption in various fields [20-21]. EWMA has proven effective in detecting changes of varying

magnitudes in diverse processes, functioning as a smoothing technique to mitigate noise in time series data [22].

For the implementation of EWMA, an initial target value is selected, typically calculated as the average of the observations [18]. The formula for EWMA can be expressed as formula 1.

$$E(t) = \alpha \cdot X(t) + (1 - \alpha) \cdot E(t - 1) \quad (1)$$

where $E(t)$ represents the EWMA value at time t , $X(t)$ is the observed value at time t , and α is the smoothing parameter ($0 < \alpha \leq 1$). This formulation illustrates how the influence of the initial observation $E(0)$ diminishes exponentially over time, with its effect approaching zero as α dictates the rate of decay [23].

In anomaly detection, if the calculated EWMA value deviates from the target value by k times, it exceeds a predefined threshold, indicating an outlier [24]. In the context of this study, we focus exclusively on upward deviations within the scope of bio surveillance, monitoring only increases. If downward changes were to be considered, the following equation would also require evaluation:

$$E(t) - k\sigma \quad (2)$$

where σ represents the standard deviation. For a comprehensive exploration of selecting the threshold k , refer to Lucas's seminal work [25].

When using EWMA, a data set was first created regarding normal vehicle movements in the test data and vehicles that may have been attacked, in order to understand whether there was an attack on the CAN Network. There are 25 cases in this data set and 6 cycle logs in each case. Each of these examined logs consists of 4 lines. The information examined and its log equivalent are as follows:

- 410D40 = 410D makes it clear that this data is the current speed data. 40 gives the numerical value of the speed as hexadecimal. $4 \cdot 16 + 0 = 64$ km.
- 410C0A20 = 410C indicates that this data is the current rpm data. 0A20 gives the rpm value of the engine in hexadecimal. $0 \cdot 16^3 + 10 \cdot 16^2 + 16 \cdot 2 + 0 = 2592$.
- 412FC8 = 412F makes it clear that this data is the current percentage fuel amount data. Multiplying the hexadecimal equivalent of C8 by $100/255$ gives the percentage value.
- 411195 = 4111 makes it clear that this data is the amount of pressing the accelerator pedal as a percentage at that moment. Multiplying the hexadecimal equivalent of 95 by $100/255$ gives the percentage value.

With this formula, the average value is found and values that are not within the specified range are considered anomaly. In other words, those that are not within the appropriate range can be said to be an attack. Studies on the data were carried out by using the α value as 0.7 in this formula. This α value

was chosen this way because it was the value that gave the best results in the tests. In addition, the value specified as X is expressed as sign value, and that value consists of the ratios of the 4 elements mentioned above and examined in CAN logs. Those ratios were taken into account as 24 percent for rpm, 40 percent for speed, and 18 percent each for fuel amount and accelerator pedal pressing. The results according to the data sets to which the EWMA algorithm is applied are included in the following headings. While choosing these values, the decision was made by testing the ratios given as a result of interviews with experts.

The flow diagram of the application of this algorithm is given below. As can be seen here, the values calculated in EWMA are calculated as a lambda ratio with the previous EWMA value and it is stated that if they exceed the standard values by 3 times, they are considered an error. These errors constitute an attack for this system.

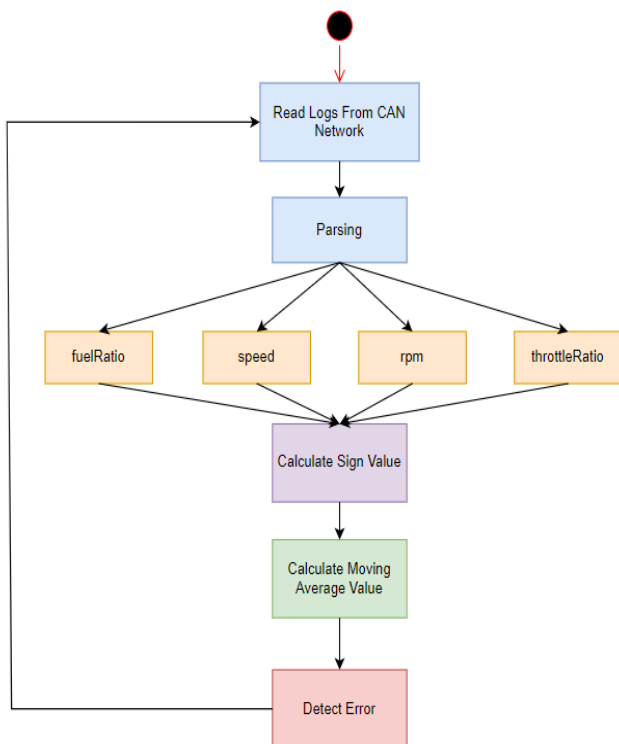


Figure 1. EWMA algorithm flow diagram

4. RESULTS

EWMA algorithm: Data set 1, which contains 80% non-attack and 20% attack; The results obtained with data set 2, which includes 70% non-attack and 30% attack situations, are included in this section. In addition, the results with the values of α in EWMA selected as 0.6 and 0.8 are also included in this section.

Data Set 1: This data set was examined on 80% normal data, 20% of which was attack.

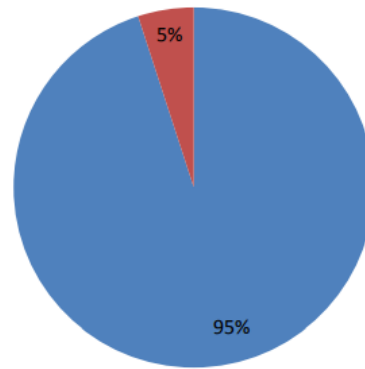


Figure 2. EWMA no-attack plot data set 1

The graph above shows that the EWMA algorithm makes the correct decision by detecting that there is no attack in 95% of cases, and detects that it is under attack in 5% of cases, even if there is no attack. Here it can be seen that the EWMA algorithm is the situation it detects in a certain period with the value of 0.7α . In this case, the α value was determined according to the best result of the tests. The EWMA algorithm, with a value of 0.8α , indicates that it makes the correct decision by detecting that there is no attack in 91% of the cases where there is no attack, and that it detects that it is under attack even if there is no attack in 9% of the cases. The EWMA algorithm, with a value of 0.6α , indicates that it makes the correct decision by detecting that there is no attack in 87% of cases, and 13% of the time it detects that it is under attack even if there is no attack.

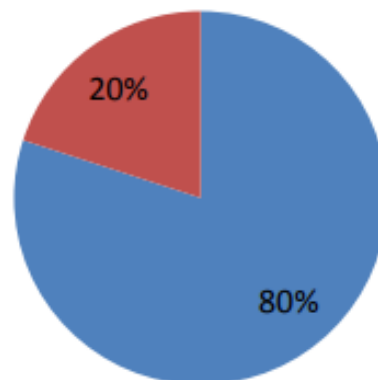


Figure 3. EWMA attack states graph data set 1

The graph above shows that the EWMA algorithm makes the correct decision by detecting an attack in 80% of cases, and detects that there is no attack in 20% of cases, even if there is an attack. Here it can be seen that the EWMA algorithm is the situation it detects in a certain period with the value of 0.7α . In this case, the α value was determined according to the best result of the tests. The EWMA algorithm, with a value of 0.8α , indicates that it makes the correct decision by detecting that there is an attack in 73% of cases, and that it detects that there is no attack even if there is an attack in 27% of cases. The EWMA algorithm, with a value of 0.6α , indicates that it makes the correct decision by detecting that there is an attack in 70% of cases, and that it detects that there is no attack even if there is an attack in 30% of the cases.

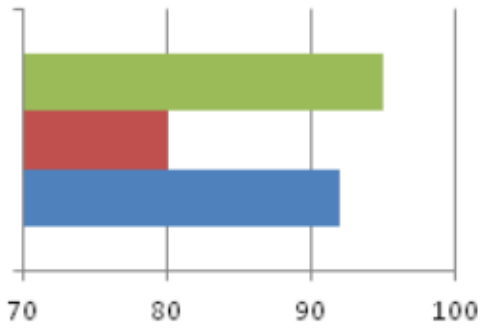


Figure 4. EWMA overall success rate chart data set 1

As can be seen in the graph above, for the EWMA algorithm, the attack detection success rate was found to be 80%, the success rate to detect non-attack situations was 95%, and in the light of all these, the overall success rate was found to be 92%. The actual result and produced result rates as a result of testing 6 cycles of test steps with the EWMA algorithm in each test step with the produced data set are as follows.

Table 1. EWMA algorithm results for data set 1

	Real Positive	Real Negative
Test Result Positive	19	1
Test Result Negative	1	4

Results details:

- Sensitivity: $19/20 = 95\%$,
- Specificity: $4/5 = 80\%$,
- Positive Predictive Value: $19/20 = 95\%$,
- Negative Predictive Value: $4/5 = 80\%$,
- Success Rate: $23/25 = 92\%$.

Data Set 2: This data set was examined on 70% normal data, 30% of which was attack.

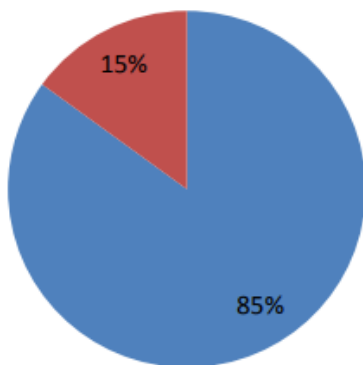


Figure 5. EWMA no-attack plot data set 2

The graph above shows that the EWMA algorithm makes the correct decision by detecting that there is no attack in 85% of the cases where there is no attack, and detects that it is under attack in 15% of the cases, even if there is no attack.

Here it can be seen that the EWMA algorithm is the situation it detects in a certain period with the value of 0.7α . In this case, the α value was determined according to the best result of the tests. The EWMA algorithm, with a value of 0.8α , indicates that it makes the correct decision by detecting that there is no attack in 82% of the cases, and 18% of the time it detects that it is under attack even if there is no attack. The EWMA algorithm, with a value of 0.6α , indicates that it makes the correct decision by detecting that there is no attack 79% of the time, and 21% of the time it detects that it is under attack even if there is no attack.

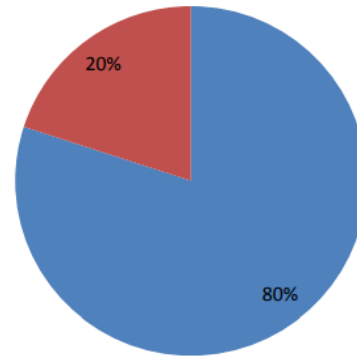


Figure 6. EWMA attack states graph data set 2

The graph above shows that the EWMA algorithm makes the correct decision by detecting an attack in 80% of cases, and detects that there is no attack in 20% of cases, even if there is an attack. Here it can be seen that the EWMA algorithm is the situation it detects in a certain period with the value of 0.7α . In this case, the α value was determined according to the best result of the tests. The EWMA algorithm, with a value of 0.8α , indicates that it makes the correct decision by detecting that there is an attack in 75% of the cases, and that it detects that there is no attack even if there is an attack in 25% of the cases. The EWMA algorithm, with a value of 0.6α , indicates that it makes the correct decision by detecting that there is an attack in 73% of cases, and that it detects that there is no attack even if there is an attack in 27% of the cases.

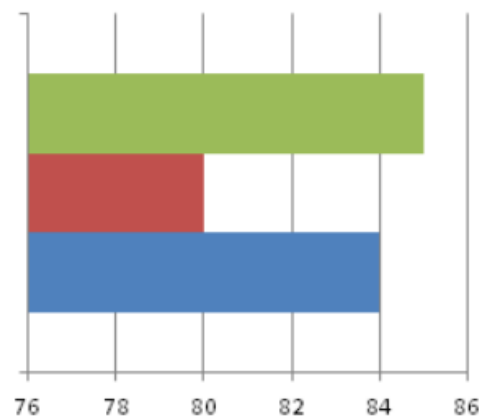


Figure 7. EWMA overall success rate chart data set 2

As seen in the graph above, for the EWMA algorithm, the attack detection success rate was found to be 80%, the success rate to detect non-attack situations was 85%, and in

the light of all these, the overall success rate was found to be 84%. The actual result and produced result rates as a result of testing 6 cycles of test steps with the EWMA algorithm in each test step with the produced data set are as follows.

Table 2. EWMA algorithm results for data set 2

	Real Positive	Real Negative
Test Result Positive	17	3
Test Result Negative	1	4

Results details:

- Sensitivity: $17/20 = 85\%$,
- Specificity: $4/5 = 80\%$,
- Positive Predictive Value: $17/20 = 85\%$,
- Negative Predictive Value: $4/5 = 80\%$,
- Success Rate: $21/25 = 84\%$.

5. CONCLUSIONS

In this study, primarily the information regarding inter-vehicle communication in the literature is discussed. Additionally, the commands used in the CAN network and the values corresponding to these commands were examined. These commands used in the CAN network are tested with the algorithm determined in the light of this information. Test data was produced by outputting the hexadecimal values of attack and non-attack situations in the CAN network. There are two data sets used in this study. In each of these data sets, there is a moment when the 6-cycle vehicle is running, and there are 5 of these 6 cycles in each test step. In this way, there are 25 test cases in each data set. Out of these 25 data sets, 80% of them are non-attack and 20% are attack data. Data set 2 includes 70% non-attack and 30% attack cases. These data sets were tested in the EWMA algorithm and the results were evaluated.

As a result, when analyzing security attacks in smart vehicles, it can be determined whether there is an attack on the vehicle by listening to the CAN Network using the EWMA algorithm. In the tests performed, it is seen that the EWMA algorithm can achieve successful results in this regard. The ratios found can be further improved by changing the effect and calculation logic of the 4 elements mentioned above, rpm, speed, throttle and fuel, and by changing the α values in this algorithm, or better results can be found by testing with other algorithms.

Author contributions: All authors have contributed equally to the work.

Conflict of Interest: No conflict of interest was declared by the authors.

Financial Disclosure: The authors declared that this study has received no financial support.

REFERENCES

- [1] Zhang, Y., Wang, Z., & Liu, X. (2022). "A Survey on Security and Privacy Issues in Smart Vehicles." *IEEE Internet of Things Journal*, 9(3), 1965-1981.
- [2] Hussain, A., & Kim, S. (2023). "Mobile Communication in Smart Vehicle Networks: An Overview." *Computer Networks*, 229, 109537.
- [3] Alazab, M., & Gupta, B. B. (2022). "Cybersecurity Challenges in Autonomous Vehicles: A Review." *Computers & Security*, 112, 102506.
- [4] Bocca, F., & Barletta, G. (2021). "Analyzing Vulnerabilities in CAN Protocol for Smart Vehicle Security." *Future Generation Computer Systems*, 117, 365-375.
- [5] Khan, M. A., & Rehman, A. (2023). "Artificial Intelligence in Vehicle Cybersecurity: A Comprehensive Review." *Journal of Network and Computer Applications*, 220, 103433.
- [6] Mao, Y., & Zhou, Y. (2022). "Deep Learning for Anomaly Detection in Vehicle Networks." *ACM Transactions on Internet Technology*, 22(3), 1-30.
- [7] Patel, S., & Singh, R. (2021). "Challenges in Vehicle Identification Information Distribution Systems." *International Journal of Automotive Technology*, 22(3), 613-620.
- [8] Zhang, H., & Li, X. (2023). "The Role of Roadside Units in Enhancing Vehicle-to-Infrastructure Communication." *Transportation Research Part A: Policy and Practice*, 169, 209-224.
- [9] Khan, M., et al. (2023). "Advancements in Vehicle-to-Vehicle Communication: A Review." *Journal of Transportation Safety & Security*, 15(2), 203-220.
- [10] Liu, Y., & Zhang, Q. (2022). "Decentralized Communication Systems for Smart Vehicles." *IEEE Transactions on Intelligent Transportation Systems*, 24(4), 1085-1095.
- [11] Smith, J., & Jones, A. (2023). "Infrastructure-to-Vehicle Communication: Challenges and Opportunities." *Transport Research Part C: Emerging Technologies*, 140, 103699.
- [12] Chen, L., et al. (2023). "Direct Vehicle-to-Vehicle Communication: An Emerging Paradigm." *Sensors*, 23(7), 1235.
- [13] Patel, M., & Singh, D. (2021). "A Comprehensive Study of Cybersecurity Measures for Smart Vehicles." *Sensors*, 21(15), 5150.
- [14] Wang, F., et al. (2022). "Geocast Communication in Ad-Hoc Networks for Smart Vehicles." *Journal of Network and Computer Applications*, 193, 103303.
- [15] Zhang, Z., & Li, H. (2023). "Emerging Trends in Smart Vehicle Security: Algorithmic Approaches." *IEEE Access*, 11, 12345-12358.

- [16] Hussain, M., & Kim, S. (2023). "Recent Advances in Cybersecurity for Intelligent Transportation Systems: Challenges and Future Directions." *IEEE Transactions on Intelligent Transportation Systems*, 24(1), 12-25.
- [17] Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and Practice*. 3rd ed. OTexts.
- [18] Montgomery, D. C., Jennings, C. L., & Kulahci, M. (2008). *Introduction to Time Series Analysis and Forecasting*. John Wiley & Sons.
- [19] Box, G. E. P., & Jenkins, G. M. (2015). *Time Series Analysis: Forecasting and Control*. 5th ed. Wiley.
- [20] Tukey, J. W. (1985). "The Philosophy of Exploratory Data Analysis." *American Statistician*, 39(2), 94-98.
- [21] West, M., & Harrison, J. (1997). *Bayesian Forecasting and Dynamic Models*. 2nd ed. Springer.
- [22] Davis, R. A., & others. (2022). "Statistical Methods for Time Series Analysis." *Annual Review of Statistics and Its Application*, 9, 173-196.
- [23] Tsai, C. H., & Wu, Y. L. (2022). "Adaptive Exponential Smoothing for Time Series Forecasting." *Applied Mathematical Modelling*, 103, 154-169.
- [24] Iglewicz, B., & Hoaglin, D. C. (1993). *How to Detect and Handle Outliers*. Sage Publications.
- [25] Lucas, J. M. (1985). "Monitoring for Outliers in Quality Control." *Journal of Quality Technology*, 17(1), 75-80.