Time Series Mean Normalization for Enhanced Feature Extraction in In-Vehicle Network Intrusion Detection System

Yusupov Kamronbek[1], Islam Md Rezanur[1], Insu Oh[2], and Kangbin Yim[2(✉)]

[1] Department of Software Convergence, Soonchunhyang University, Asan, Korea
{yuskamron,arupreza}@sch.ac.kr
[2] Department of Information Security Engineering, Soonchunhyang University, Asan, Korea
{catalyst32,yim}@sch.ac.kr

Abstract. The growth of electronic control units (ECUs) has led to modern automobiles being more convenient and technologically advanced. In order to communicate amongst these control units, the widely used Controller Area Network (CAN) communication protocol is used. CAN buses do not, however, come with built-in security features while being reliable and cost-effective. Because of this weakness, they are vulnerable to several attacks from prospective enemies. The installation of an Intrusion Detection System (IDS) is, nonetheless, a practical answer to this issue. A CAN bus system's security can be considerably improved by the use of an IDS. We suggest a Long Short-Term Memory (LSTM) based intrusion detection system that is capable of effectively addressing the detection and security needs of CAN bus systems. We think that using Long Short-Term Memory (LSTM) models has a number of benefits, including highly accurate classification and effective anomaly detection. Our research findings have shown that our Long Short-Term Memory (LSTM) based IDS successfully identifies assaults on CAN systems with an accuracy rate of 99% and little loss. By providing this solution, we want to aid in the creation of reliable intrusion detection systems that can successfully protect CAN bus systems.

1 Introduction

A variety of tools and technology have helped the contemporary automobile sector advance significantly over time. ECUs, or electronic control units, are a possible addition to this list. When compared to today, modern cars can include anywhere between 50 and 100 ECUs [1]. Previously, cars had significantly less of this type of electrical technology. They facilitate driving, aid with control, and enhance comfort.

The CAN (Controller Area Network) bus is used by these integrated into the body of the vehicle components to transmit and receive crucial data. Due to its affordable, effective, and dependable design, the CAN bus is a standard communication protocol that is widely used in the automobile sector [2]. These innovations are making automobiles smarter and more interconnected. But in addition to the comfort and connection, the CAN bus has developed into a target for hackers. It is susceptible to possible incursions

L. Barolli (Ed.): BWCCA 2023, LNDECT 186, pp. 302–311, 2024.
https://doi.org/10.1007/978-3-031-46784-4_29

because of its flexible structure and simplistic design. Modern vehicles are a target for attackers because as they become increasingly interconnected with the outside world, new potential for hacks opens. The CAN bus contains built-in weaknesses such as lax access control, insufficient encryption, and inadequate authentication. By taking advantage of these weaknesses, attackers may take control of ECUs and use them to send malicious messages across the internal network of the car. The driver's and passengers' safety might be seriously jeopardized as a result of severe repercussions such as unlawful control of the vehicle, abrupt braking, or engine shutdown. It is crucial to install an intrusion detection system (IDS) in automobiles to reduce these hazards and avoid such accidents. An IDS enables the detection and prevention of several assaults, both locally and remotely from the vehicle. It monitors CAN bus data integrity, analyzes network traffic, spots suspicious behavior, and reacts to threats immediately. Vehicles may be protected and possible dangers from auto break-ins can be reduced by adopting an IDS.

This article focuses on an intrusion detection system that uses LSTM (Long Short-Term Memory), which is an anomaly detection technique relevant to automotive systems. A recurrent neural network with long-term memory that can handle sequential data is called an LSTM. The LSTM model has benefits including the ability to handle sequential data, retain knowledge over time, draw conclusions from past data, and adjust to changing environmental conditions. These features enable the real-time identification of assaults or anomalous activity on the CAN bus as well as the detection of novel attack types without the need for tight rules. The article also looks at other intrusion detection techniques for the CAN network, which could produce better outcomes but have drawbacks including a larger chance of false positives. These techniques may be quite sensitive, which might result in false positives and add to the stress on the system.

In this work, we provide a powerful LSTM-based intrusion detection technique for automobile CAN bus systems, focusing on attacks like DoS, Fuzz, Malfunction, and Replay. The paper is divided into many sections, including related work (Sect. 2), a description of the CAN protocol and its structure, as well as attack dataset (Sect. 3), LSTM, feature extraction and data pre-processing (Sect. 4), Evaluation performance and results (Sect. 5), and a commentary (Sect. 6) that wraps up the subject.

2 Related Work

The protection of the CAN system against intrusions and attacks is a top priority when it comes to system security in the field of mechanical engineering. Hackers who take over the CAN system might modify data, putting the driver's and passengers' safety in peril and having serious repercussions. In order to maintain the safety and security of CAN systems, it is essential to deploy an intrusion detection system (IDS). Modern attack classification and intrusion detection methods have been suggested by researchers.

In one research, Md. Delwar et al. investigated the effectiveness of an IDS model based on convolutional neural networks (CNN) to detect network assaults [3]. They made use of data from real Toyota, Subaru, and Suzuki cars that have been the targets of DoS, Fuzz, RPM, and Gear Spoofing assaults. Their model performed exceptionally well in classifying assaults across various vehicle kinds, obtaining a 99.99% detection accuracy. Markus et al. presented CANet, a noteworthy additional model [4]. This unsupervised model performed exceptionally well at picking up signal alterations that are

difficult to pick up using more traditional methods. Using both actual and synthetic data, it effectively recognized several unidentified assaults, achieving a high negative rate (0.99). In order to identify attacks, Abdul et al. developed CANintelliIDS [5], a technique based on deep recurrent network units (GRUs) and convolutional neural networks (CNNs). Their research revealed an F1 score for class 0 (normal) of 98.21% and class 1 (attack) of 97.38%. Hyun et al. carried out three different sorts of assaults using CAN communications from several known automobiles [6]. They discovered that the interval between CAN broadcasts was a key factor in the detection of attacks. Across all experiments, their suggested IDS had a 100% detection accuracy rate with no false positives. Min et al. suggested a successful deep neural network (DNN)-based intrusion detection method to protect the CAN system [7]. Their system allowed the IDS to recognize any hacker assault on a vehicle by classifying packets into categories that were benign and malicious. The method has a 98% detection success rate. Additionally, Md. Delwar et al. created an IDS model based on LSTM [8] to recognize DoS, Fuzz, and spoofing threats. Their model correctly identified and classified anomalies with 99% accuracy and few false positives. They experimented with several LSTM setups to obtain excellent detection speed and accuracy. For clustering assaults and routine data in a CAN system, Harini et al. developed an IDS model based on unsupervised learning. They trained patterns for both CAN classes using a two-phase methodology that included an autoencoder and a mixed Gaussian mixture model (GMM) [9]. Using this method, it was possible to recognize CAN system assaults. Thien et al. offered the ResNet-based CANPerFL architecture and provided an IDS for binary encoding CAN ID categorization [10]. The temporal and geographical organization of the CAN ID sequence was represented by a matrix created by this method. The accuracy attained with this method was around 99%. Heatmaps were utilized as input for a number of models, such as VGG-16, AlexNet, and ResNet-50, in the universal intrusion detection system developed by Md Rezanur et al. The CAN ID, time interval, and Hamming distance were used to create these heatmaps [11]. By attaining a 99% accuracy rate, the ResNet-50 model surpassed the competition.

3 CAN Bus and Attack Datasets

3.1 Controller Area Network (CAN)

Controller Area Network is a well-known communication protocol [12] that is widely used in the mechanical engineering field. Using this communication protocol, communication is established between Electronic Control Units (ECU) and other vehicle systems that are located inside the vehicle, this communication method is considered reliable and efficient. Convenient architecture allows you to interact with different devices through one channel, which makes it convenient to use. Basically, CAN is used to establish communication between ECUs. The CAN frame format, as shown in Fig. 1, includes several parts. Such as Start of Frame (SOF) which demonstrates the start of a message. After SOF demonstrated the CAN ID is displayed which is the next part of the CAN frame. This part helps to identify the sender, the ECU which is responsible for a certain element. The standard CAN arbitration field is 11 bits in size, and the extended form is 29 bits. Remote Transmission Request (RTR) is part of a Data Frame that is used to transmit data to the network by another node. By using RTR, you can avoid aggregate

network traffic. Data Length Code (DLC) shows the number of bytes of data contained in a message. After it there is a Data Field which has useful information depending on what information is being transmitted. Cyclic Redundancy Check (CRC) is responsible for error detection. The last part is End of Frame (EOF).

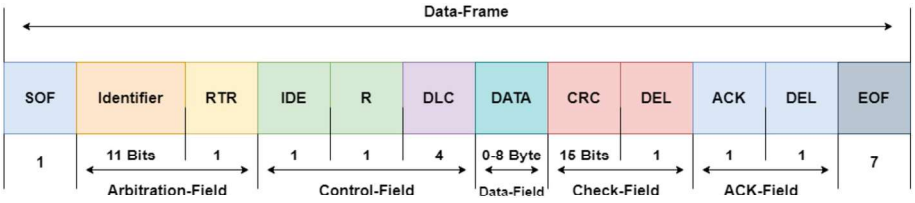

Fig. 1. Standard Data-Frame CAN [12]

3.2 Denial of Service (DoS)

Denial of Service (DoS) [13] attacks are nefarious attempts by cybercriminals to obstruct a system's regular operation, making it difficult or impossible for users to access or utilize the system. A DoS attack on the CAN bus seeks to stop the system from processing or transmitting legitimate instructions and messages, disrupting its normal operation. If a DoS attack is successful against the CAN bus, there may be serious repercussions, including engine or brake system failures that might result in accidents. In our situation, the DoS attack was executed by periodically injecting messages into the system with the CAN ID "0000" at intervals of 0.3 and 0.5 ms. The most typical identifier employed in this kind of attack is represented by the selected CAN ID "0000". The attack's goal is to flood the CAN bus with many messages bearing the targeted ID, overloading the system and resulting in a denial of service.

3.3 Fuzzing

A fuzz attack, sometimes referred to as fuzzing, is used to test a system's robustness by flooding it with a lot of false, inconsistent, or unexpected input [13]. Hackers can create fuzzing tools for the CAN bus that produce false and random messages while adhering to the CAN protocol. We used a fuzzing attack on our dataset, delivering messages with CAN ID and DATA values that were fully random at intervals of 0.3 and 0.5 ms. A fuzzing attack aims to cause ambiguity and confusion on the CAN bus. The attack seeks to prevent the system from operating normally by delivering messages with arbitrary identities and contents. This attack may result in unpredictable behavior since automotive systems are often not built to manage such random and unexpected input. This unpredictability may reveal weaknesses or result in system failures in the automobile, leaving them open to unanticipated problems.

3.4 Replay

Replay attacks are a particular kind of attack that hackers employ to target a system. A signal supplied to the CAN [14] communication system is intercepted and retransmitted in this attack by the attacker. To attack the CAN bus, perform the following steps: In order to intercept and record messages, the hacker first employs specialized equipment. The hacker then picks the messages that can be relayed back to the CAN system after sorting through the captured conversations. Following that, the target messages are picked once the chosen messages have been processed. Finally, these chosen messages are sent again, sometimes with their original sender names and contents, to deceive the system into considering them to be legitimate data and acting accordingly. The hacker inserted new traffic onto the CAN bus during the replay attack by removing normal traffic that normally occurs at a certain moment. CAN communication was seen during the assault via an actual cars OBD-II connector. In order to build the datasets used in the attack, messages were added. Each incident of message injection lasted between three and five seconds, and the captured data included information on 300 such occasions. Each dataset has between 30- and 40 min worth of CAN traffic in total.

3.5 Malfunction

Malfunction is one type of cyber-attack that is used to cause crashes or unexpected system behavior. In our case, the Malfunction attack can cause attacks on the CAN system with the aim of interfering with the normal operation of the communication. This attack is made by introducing incorrect or dangerous signals associated with the CAN [15]. Using flaws in the CAN protocol, attackers can control the ECU or monitor sensors and can change the messages that are used in vehicles. For our data set, the attack was carried out in the following order: a specific CAN ID was chosen, which is used in KIA SOUL CAN ID 0x153. The main goal was to change the values in the 8-byte data field. In order to carry out the attack, the number 00 and random numbers were discovered. Another injection was done by randomly choosing a CAN ID and combining the result with data fields. As a result of the attack, the car showed non-standard reactions after the injection, which eventually led to failures. The injections were given at a time interval of 5 s every 20 s.

4 Mean Normalization and Time Series for LSTM

4.1 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) that was developed to eliminate the vanishing gradient problem. In simple words, LSTM is an extended version of RNN that copes well with input data using sequential dependencies. The strengths of this model are it collects and can use different time stages. With the help of memory cells, LSTM models can recall or forget data during processing. In order to control information later, there are cell gates; the forgetting gate decides which information should be skipped further and which should be forgotten. The update gate decides which data needs to be changed inside the cell and the input gate decides which

information needs to be entered into the cell. With this method, all layers affect the result of the network. Thanks to this structure, LSTM can effectively cope with long-term dependencies. LSTM models are good at prediction and classification tasks (Fig. 2).

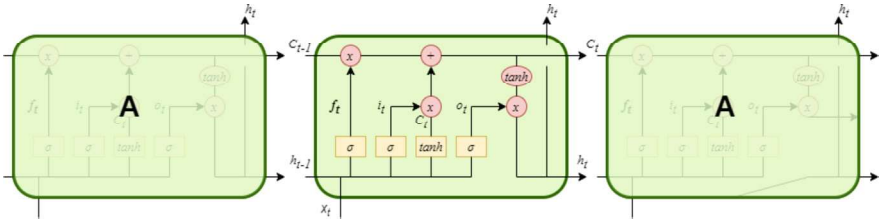

Fig. 2. Long Short-Term Memory (LSTM) structure

4.2 Feature Extraction and Dataset Pre-Processing

Our data set was collected from a real KIA SOUL machine and includes attacks such as DoS, Fuzz, Replay, Malfunction and Attack Free legitimate data [17]. At the beginning of our experiment, we had to clean and check the data for the content of incorrect data. After cleaning the data, we had to separate the attacked data from the legitimate data. Our set included 15 columns showing different information that was transferred between different devices. These columns contained information such as time since start, CAN ID, Time_gap which shows the time interval between CAN messages, Data Length Code (DLC), Payload in hex format, and 8 columns that show one each of Payload 1–8. In addition, in our dataset there is a marking of messages "R" and "T" which denote messages attacked or legitimate. We have added a new column to our dataset to convert CAN ID to categorical data to avoid the issue of huge CAN ID numbers during model training. However, many researchers have done research and they used all the columns to train their model. In our experiment, we only focused on two columns with which to classify attacks and detect them. We only used Categorical_CAN_ID and Time_gap. After fetching the desired columns, we connected all the datasets that we had, which is, DoS, Fuzz, Malfunction, Replay, Attack Free and then we labeled the data differently to distinguish them by the number we assigned them. We marked the data with numbers as follows: legitimate data was assigned the number 0, DoS - 1, Fuzz -2, Malfunction - 3 and Replay - 4. After combining all the data sets into one, we started pre-processing, which consisted of two stages: Data normalization and Data scaling. To make sure that all data points are on the same scale, we used "Mean Normalization", this method scales the data depending on the average and standard deviation. Our data preprocessing determines the mean and standard deviation for each dataset. We have changed the numbers to fit a certain range. Our function is used as follows ($x\_normalized = (x - \mu)/\sigma$). x_normalized shows the normalized value of x, where x represents the original number. This way you can get the average of the data. In the second step, we used data scaling, which is done by transforming the values so that they are in a certain range. To do this, we used the formula (array-mean)/std to scale the data. After that, all the data that has passed through this

function becomes scaled and zero-aligned. After pre-processing, we divided the datasets into small chunks in order to effectively regulate them. After that, we mix these segments and combine them again, after which we get the moved dataset. This method ensures that the data is evenly distributed and moved using this method, you can avoid overfitting the model and avoid any dependencies. Lastly, we used a time series that was organized with time step values of 10. To predict the next value, we developed a sliding eye method that takes as input data set equal to 10 of the time steps. The goal was to make a window of 10 and the model based on 10 messages was to predict the next value. The window is then moved one step down after the prediction and this cycle is repeated until the end of the data set. Then our dataset was split into two parts for training the model and for testing the LSTM. Each component has been converted to a NumPy array, and because our model is working with 3D data, we have changed the format of the data.

5 Evaluation Performance

We created an LSTM model using the TensorFlow library (version 2.11.0). Our model's architecture consists of four LSTM layers, each with 128 units. Each layer of the model was coupled to a batch normalization layer and a dropout layer with a dropout rate of 0.2. The hyperbolic tangent (tanh) is the activation function employed in the LSTM layers. A dense layer with five units and a Softmax activation function makes up the model's top layer. Using the above parameters, we trained the model using the backpropagation approach and the RMSprop optimizer. We shuffled the training data before to each epoch to improve the generalizability of the model. We used a variety of metrics and measurements to assess the model's efficacy. The percentage of samples that were properly categorized served as our main criterion for accuracy. The difference between the predicted and actual class labels was also measured using a loss estimator. We employed Early Stopping, a standard model training strategy, to reduce overfitting and enhance learning. When the chosen measure stops improving after a predetermined number of epochs, Early Stopping aims to end model training. It also stores and retrieves the model that produced the greatest outcomes with the least amount of loss. With a patience of 10 epochs and the recovery of the best weights, we used Early Stopping in our experiment. In order to constitute an improvement, a minimum change (min delta) of 0.0001 has to occur. In Table 1 and Fig. 3. The experiment's results showed that we were able to attain an accuracy of 98.87% and a loss coefficient of 0.0300 on the training sample. At epoch 97 out of 150, when this conclusion was reached, the early stopping parameters stopped the training. On the test sample, we had 99% accuracy. These findings show that the model is highly generalizable and capable of accurately classifying fresh data. We created two models for data categorization in this experiment. Data might be divided into five categories using the basic model: normal, dos, fuzzing, replay, and malfunction. For the second model, just the normal and attacked data classes were created, using BinaryCrossEntropy. The results of the second experiment showed that our binary model maintained the same high accuracy of 99% after 50 epochs with a loss of 0.0298 as shown in Table 2.

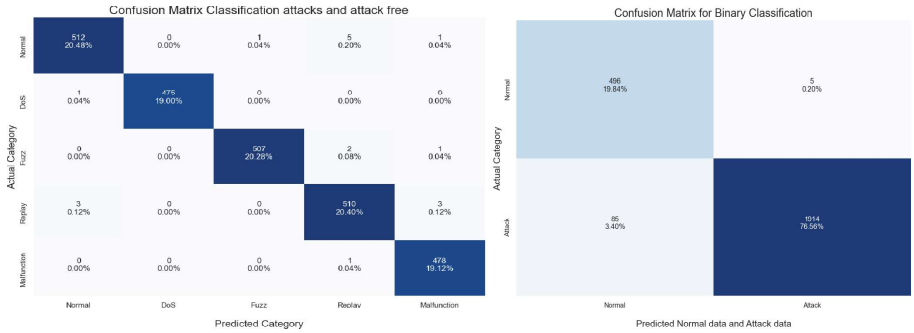

Fig. 3. Confusion Matrix for Multi-class and Binary Classification

Table 1. LSTM model report for Multi-class Classification.

	precision	recall	F1-score	support
Normal	0.99	0.99	0.99	519
DoS	1.00	1.00	1.00	476
Fuzz	1.00	1.00	1.00	510
Replay	0.98	0.99	0.99	516
Malfunction	0.99	0.99	0.99	479
Accuracy			0.99	2500
Macro avg	0.99	0.99	0.99	2500
Weighted avg	0.99	0.99	0.99	2500

Table 2. LSTM model report for Binary Classification.

	precision	recall	F1-score	support
Normal	0.97	0.99	0.98	484
Attack	1.00	0.99	1.00	2016
Accuracy			0.99	2500
Macro avg	0.99	0.99	0.99	2500
Weighted avg	0.99	0.99	0.99	2500

6 Conclusion

In the current experimental work, two LSTM models that can classify attacks and legitimate data were built in order to implement LSTM models on CAN bus Intrusion Detection Systems (IDS). The goal was to develop an effective intrusion detection system that can detect and classify attacks with high accuracy. In the initial experiment, we built an

LSTM model that can classify 4 types of attacks (DoS, Fuzz, Malfunction, Replay) and recognize legitimate data. Our initial experiment results showed that our model was able to classify attacks and anomaly detections with an accuracy of 99%. The results demonstrate that our model can be used as an effective intrusion detection system (IDS) for a CAN system. In the second experiment, we built an LSTM with binary classification, that is, to distinguish only two classes of attacks or legitimate data. The results of this experiment also showed much more promising accuracy with 99%. The results show that both models perform well in attack detection and classification tasks. Our future research will focus on extending and improving current models using different data preprocessing techniques and additional approaches. We expect that using these methods you can get much better results and reduce the risk that may arise from attackers. We think our results can help researchers create effective protection for vehicles.

Acknowledgments. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2021R1A4A2001810) and Institute for Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2022-0-01197, Convergence security core talent training business(SoonChunHyangUniversity).

References

1. ECU: https://www.linkedin.com/pulse/20140626152045-3625632-car-software-100m-lines-of-code-and-counting
2. Bozdal, M., Samie, M., Aslam, S., Jennions, I.: Evaluation of can bus security challenges. Sensors **20**(8), 2364 (2020)
3. Hossain, M.D., Inoue, H., Ochiai, H., Fall, D., Kadobayashi, Y.: An effective in-vehicle CAN bus intrusion detection system using CNN deep learning approach. In: GLOBECOM 2020–2020 IEEE Global Communications Conference, pp. 1–6. IEEE (2020)
4. Hanselmann, M., Strauss, T., Dormann, K., Ulmer, H.: CANet: an unsupervised intrusion detection system for high dimensional CAN bus data. IEEE Access **8**, 58194–58205 (2020)
5. Javed, A.R., Ur Rehman, S., Khan, M.U., Alazab, M., Reddy, T.: CANintelliIDS: Detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU. IEEE Trans. Netw. Sci. Eng. **8**(2), 1456–1466 (2021)
6. Song, H.M., Kim, H.R., Kim, H.K.: Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network. In: 2016 International Conference on Information Networking (ICOIN), pp. 63–68. IEEE (2016)
7. Kang, M.J., Kang, J.W.: Intrusion detection system using deep neural network for in-vehicle network security. PLoS ONE **11**(6), e0155781 (2016)
8. Hossain, M.D., Inoue, H., Ochiai, H., Fall, D., Kadobayashi, Y.: LSTM-based intrusion detection system for in-vehicle can bus communications. IEEE Access **8**, 185489–185502 (2020)
9. Narasimhan, H., Vinayakumar, R., Mohammad, N.: Unsupervised deep learning approach for in-vehicle intrusion detection system. IEEE Consum. Electron. Mag. **12**, 103–108 (2021)
10. Hoang, T.N., Islam, M.R., Yim, K., Kim, D.: CANPerFL: improve in-vehicle intrusion detection performance by sharing knowledge. Appl. Sci. **13**(11), 6369 (2023)

11. Islam, M.R., Oh, I., Yim, K.: Universal intrusion detection system on in-vehicle network. In: Barolli, L. (ed.) Innovative Mobile and Internet Services in Ubiquitous Computing: Proceedings of the 17th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2023), pp. 78–85. Springer Nature Switzerland, Cham (2023). https://doi.org/10.1007/978-3-031-35836-4_9
12. Hpl, S.: Introduction to the controller area network (CAN). Application Report SLOA101, pp. 1–17 (2002)
13. Lee, H., Jeong, S.H., Kim, H.K.: OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame. In: 2017 15th Annual Conference on Privacy, Security and Trust (PST), pp. 57–5709. IEEE (2017)
14. Kang, H., Kwak, B. I., Lee, Y. H., Lee, H., Lee, H., Kim, H. K.: Car hacking and defense competition on in-vehicle network. In: Workshop on Automotive and Autonomous Vehicle Security (AutoSec), vol. 2021, p. 25 (2021)
15. Han, M.L., Kwak, B.I., Kim, H.K.: Anomaly intrusion detection method for vehicular networks based on survival analysis. Veh. Commun. **14**, 52–63 (2018)
16. Van Houdt, G., Mosquera, C., Nápoles, G.: A review on the long short-term memory model. Artif. Intell. Rev. **53**, 5929–5955 (2020)
17. Datasets: https://ocslab.hksecurity.net