Publishing Excellence
Inspiring Progress
ARISTON PUBLICATIONS

CompSci & AI Advances

ISSN: xxxx-xxxx https://aristonpubs.com/computers-ai-advances https://doi.org/10.69626/cai.2024.0074

RESEARCH ARTICLE

Adaptive AI Architectures for Autonomous Systems: A Hybrid Deep Learning Framework

S. Prabu [1,*], P. Jeevitha [2], S. Ramya [3]

ABSTRACT: Vehicles have become an intrinsic part of our lives as one of the most popular ways of private transportation. Even though it provides comfort and safety, private transportation poses road safety risks. Road fatalities are increasing due to traffic, high speed, and driver error. As a result, safety is a top priority in vehicle manufacturing and operation. The advancements in the automobile industry strive to provide increased safety benefits compared to its previous generations. Many modern vehicles include driver assistance systems that aid drivers in various ways. These systems offer helpful information about traffic, congestion levels, blockage, alternative routes to avoid congestion, etc. When a threat is detected, the driver assistance systems may take control of the vehicle from the driver and undertake simple tasks to complex manoeuvres. It also enables road safety, better driving, and reduce fatalities by limiting human error. Such vehicles incorporating the automated driving systems to communicate with the outside world are called Connected and Autonomous Vehicles (CAVs). CAV has emerged as a transformative technology in the automobile sector that has a great potential to change our daily life. Although the ever-increasing use of CAV has numerous advantages, the potential drawbacks, such as security and vulnerability to hacking, are not negligible. CAVs use a variety of sensors to build a virtual map of their surroundings to drive in the correct lane within the speed limit, avoid collisions, and detect obstacles in their immediate physical environment.

Keywords: Private Transportation; Road Safety; Traffic Fatalities; Driver Assistance Systems.

Received: 24 February 2024; Revised: 27 March 2024; Accepted: 17 April 2024; Published Online: 15 May 2024

1. INTRODUCTION

In recent years, Automobile Industries have competed to launch the first fully Autonomous Vehicle (AV). In the future, we will see a lot of selfdriving cars around the world. Many companies like Ford, Toyota, Volvo, Tesla, etc., have been taking test drives in recent years. In 2017, Ford spent $1 billion on Artificial Intelligence (AI) start-up Argo AI [1]. In 2019, Ford's Argo AI had put $15 million into forming an AV research Centre. In 2015 Toyota invested $1 billion to develop an AV. Volvo's joint venture with Uber spent $300 million to develop next-generation self-driving cars [2]. BMW with Daimler spent $250 million to work on the development of self-driving cars.

Many Automobile industries have invested in developing Avs [3]. An AV can operate itself and perform necessary functions without any human intervention through the ability to sense its surroundings. An AV utilizes a fully automated driving system to allow the vehicle to respond to external conditions that a human driver would manage. It relies on advanced AI and Machine Learning (ML) systems to understand their environment and react to commands. AVs are also known as self-driving cars, driverless cars, or robotic cars. The self-driving car is becoming a standard as these 2 technologies continue to mature. Connectivity and automation are two separate forms of technology that are often mentioned in the same breath – Connected and Autonomous Vehicles (CAVs). If these technologies can work in tandem, they might solve the problems of traffic and driver mistake, making roads safer and healthier for everyone. Many people throughout the world see self-driving

[1] Department of Electronics and Communication Engineering, Mahendra Institute of Technology, Namakkal -637503, India.
[2] Computer Science and Engineering, Hindusthan Institute of Technology, Coimbatore, India.
[3] Electronics and Communication Engineering, Sri Krishna College of Technology, Coimbatore, India .

* Author to whom correspondence should be addressed:
vsprabu4u@gmail.com (Prabu S)

vehicles as the transportation industry's next big technical leap forward. Their expected revolutionary effects on world safety include, but are not limited to, improved transportation efficiency, less congestion, and fewer accidents [4]. A long-awaited technological revolution in the transportation industry is about to begin, thanks to the improvements that have been in the works for the last several years. A lot of people's dreams will soon come true, or the world will be struck hard by an unexpected reality.

A number of CAVs have incorporated different technical developments to bring about self-driving automobiles, which will provide safer and more efficient transportation options [5]. To detect their environment, AVs use sensors placed all around the vehicle. Range measurements, lane marker identification, and road anomaly detection (e.g., holes and pits) are the primary applications of Light Detection and Ranging (LIDAR). The RADAR system uses radio waves to follow other vehicles. When an AV gets close to an obstacle, the time it takes for the radio waves to return from the obstruction to the device determines the distance, angle, and speed of the obstacle. Operating at 24, 74, 77, and 79 GHz, respectively, are short-range, medium-range, and long-range radars. Video cameras can also detect traffic signals and scan road signs. Since the data produced by each device's sensors is structured differently, sophisticated AI-powered software will subsequently analyze all of these inputs. The actuators in the cars use the processed data to accomplish things like map routes, regulate three axes of steering, braking, and accelerating, and avoid obstructions.

It is critical to check the program's integrity when updating a CAV's software. Malicious apps may be used by attackers to steal privileges or acquire access, inject malicious code into the software installed in the car, or even encourage the download of updated apps with harmful intent. Malicious software may impersonate legitimate programs while secretly collecting user input in order to steal account information, activate suspicious service ports, or save permission for future use. By establishing contact with the command and control server, such malicious software might potentially facilitate further remote assaults. Consequently, safeguarding vehicle software is of utmost importance whenever a communication channel is established between the vehicle and the surrounding infrastructure or when an external device, such a smartphone, is linked to the car via an internal interface [6].

For ECU authentication and stream authorization, demonstrated the use of an asymmetric cryptographic method based on the Advanced Encryption Standard (AES). Every message stream is allowed and the asymmetric keys for stream access are provided to the ECUs during the authentication process, which is done against a central security module utilizing stream authentication. The computational feasibility of altering or inserting data packets is rendered impossible by this study. Additional processing and data transmission time is a cost that has to be considered when building a real-time cryptography solution. While the specific delay that occurred is not disclosed by the study, they do state that the "impact of our approach is small." Despite the promising results, this study has not been integrated into any new vehicle designs at this time. Each tire has a "direct" little device called a tyre-pressure monitor system (TPMS) [7] that regularly updates the vehicle's management system with data relevant to that tire. Considering the complexity of the complete vehicle, this sensor is modest and has a rudimentary purpose. Nevertheless, it deserves examination due to current attention and privacy issues. A basic sensor's ability to affect the car and the driver in such a way is worrisome.

2. LITERATURE SURVEY

A recently study examined the present state-of-the-art defense mechanisms for Autonomous Driving mechanisms (ADSs) and a range of potential threats to these systems [8]. An exhaustive analysis of the ADS process is the first step of the study. This includes physical and cyber threats, as well as adversarial attacks on various deep learning models. Many promising areas of study have been proposed to enhance the safety of autonomous driving systems that rely on deep learning. These areas include training models to be more resilient, testing and verifying models, and detecting anomalies using cloud or edge servers. Machine learning presents a number of challenges in vehicular networks, which were thoroughly examined from the researcher study [9]. Furthermore, they showcase the CAVs' machine learning pipeline and go over a plethora of possible security issues associated with ML technology. In particular, their studies deal with adversarial ML attacks on CAVs, and they lay forth a plan to defend against these attacks in different contexts. In order to identify intelligent black hole attacks, Existing study [10] developed a system that is exclusive to autonomous and connected vehicles (ACVs). While building the system, four important factors are examined: Hop Count, Destination Sequence Number, Packet Delivery Ratio (PDR), and End-to-End delay (E2E). They tested IDBA's efficacy against AODV using the Black Hole (BAODV), Intrusion Detection System (IdsAODV), and EAODV algorithms [11]. Detailed simulation results show that the IDBA outperforms state-of-the-art methods on many key metrics, including packet loss rate, throughput, E2E, routing overhead, and DDR. Existing study [12] introduced the Targeted Attention Attack (TAA) method for actual road sign assaults. Particularly noteworthy are the following contributions: Using the soft attention map, they accomplished three things: 1) highlighted important pixels while ignoring zero-contributed areas, which helps with natural disturbance generation; 2) developed a universal attack that optimizes a single perturbation or noise using a set of training images and the attention map; and 3) made an easy-to-optimize basic objective function. Experimental results reveal that the TAA technique is superior to the well-known RP2 method in terms of attack success rate (up more than 10%) and perturbation loss (down around 25%). A novel approach to aiding a host vehicle in assessing the mobility behavior of a target vehicle and subsequently the accuracy of data transmission in cooperative vehicular communications was proposed by Existing study [13]. At first, the detecting system takes the positional data from the V2V signals it has received and uses it to mimic the target vehicle's motion behavior on the host vehicle. It also uses the unscented Kalman filter to forecast the car's future states. Whether it's from a Sybil attack or cooperation, the simulation findings demonstrate that the system can identify anomalous reports with a precision of more than 0.97.

To aid host cars and V2X edge apps in validating the legitimacy of data exchange in 5G vehicular networks, Existing study [14] proposed a possible cooperative verification approach. First, the host vehicle's detecting systems (local detector) and the RSU's (global detector) glean information about the target vehicle's status and characteristics from the CAMs that have been received. Using simulation, their study shows a significant impact on detection performance and reaction time, especially for quickly identifying Sybil and fake data assaults. An aggregate operator called the metric temporal counting quantifier was proposed by Existing study to describe a policy depending on the number of times specific sub-policies are fulfilled in a given past time period. This policy language is based on a past-time version of MTL. While not all policies can be monitored in a trace-length independent way, they did show that a large class of rules specified using the language can be and provided a specific strategy for doing so. They proved that the proposed method can successfully build and oversee quantitative rules drawn from real-world investigations by building and testing the algorithm using an existing Android monitoring framework and an AV simulation platform.

In their demonstration of a new attack vector, Existing study [15] showed how existing embedded devices may have their data stealthily leaked. The Device Tree, a data structure that describes a computer's hardware characteristics, was used to get specific details on the 35-system target. Using this knowledge, they devised a clandestine assault that transmits data straight from memory to analog peripherals in an effort to cross the air gap. The assault stays in the peripherals, undetected by the main CPU, by purposefully short-circuiting certain components. The defense was also tested for overall efficacy and performance overhead, which led to little performance cost and strong detection of the underlying threat. Existing study [16] laid the groundwork for a thorough comprehension of the good and negative uses of machine learning, and they lauded the weaknesses of ML systems in the face of traditional and ML-based attacks. Within the framework of cybersecurity and CPS, we explore the positive and negative applications of machine learning. We have now covered the dark side of machine learning, or the weaponization of machine learning, which includes increasing infiltration and obfuscation tactics, disrupting system stability and service, and violating user privacy. Recently, AVs have been troubled by data thefts and worries about sensing and tracking technologies. A secure and intelligent sensing and tracking architecture based on Blockchain was introduced by Existing study [17] for AV systems that use communication networks beyond 5G. Secure object detection and tracking via BC is guaranteed by the proposed architecture, which deploys AI algorithms at the edge servers. Any application requiring low latency, high reliability, and robust security may benefit from the proposed system, which surpasses its predecessors.

An approach to real-time data analysis based on machine learning was created by Existing study [18] to identify malicious activity in massive amounts of network traffic. They started by setting up a detection architecture that the intrusion detection module needed to identify and prevent malware from infecting the automobile using a smartphone. After that, they compared their new algorithm to the current ones, came up with a cost-effective way to detect malicious activities in a network setting, and then, tested it to make sure it was accurate. An new approach to evaluating the risks of AV accidents was proposed by Existing study [19] via the comparison to a more recognized and quantifiable risk: human behavior. Autonomous vehicle (AV) safety predictions compared to human drivers are based on this technique. Calculating the risk of an accident involving an AV may be done by comparing its behavior to that of safe human drivers. To compare the driving behaviors of humans and autonomous vehicles, Convolutional Neural Networks (CNN) are used to simulate an end-to-end AV model. An algorithm known as Gaussian Processes (GP) is used to identify contextual driving abnormalities. A risk score is then computed based on the frequency and severity of these anomalies. This paper provides a foundation by addressing the difficulties of AV risk modeling. To identify fake over-the-air software upgrades in autonomous vehicles, Existing study [20] developed an enhanced update system. The update framework's security was enhanced with the help of a Convolutional Neural Network (CNN). In a very precise manner, the suggested system can distinguish between malicious and safe software executables.

3. PROPOSED WORK

Time series classification can be done in a variety of ways. The majority of them have two main phases: the first one utilizes some method to measure the difference between time series to categorize them (dynamic time warping is a good example), and the second stage uses some tools (simple statistics, advanced mathematical methods, etc.) to depict the time series as feature vectors. An algorithm can be applied in the second stage to classify the data. It may be anything from k-nearest neighbors and Support Vector Machines (SVMs) to deep neural network models. But there is one thing that all these methods have in common: all need feature engineering as a distinct stage before classification. However, there exist models that not only include feature engineering into one framework but also remove the need for human feature engineering. They also extract features and provide meaningful time series representations autonomously. Recurrent and Convolutional Neural Networks (CNNs) are widely used models. According to research, employing

CNNs for time series classification has many significant advantages compared to other approaches. They are incredibly noise-resistant models that can extract highly informative, deep, time-independent features.

The architecture of a typical CNN is depicted in Figure 1, which is composed of a sequence of steps. SS denotes the sub-sampling of the feature map. The architecture's initial few stages are made up of two layers: convolutional and pooling. And, the final step is made up of a fully-connected layer and a standard classifier. The convolutional layers have several filters that combine the input from the preceding layer with a set of weights to create a feature output, referred to as a feature map. Neurons are linked directly to the input data points in every filter, multiplying the data points with weights. The weights of all the neurons in a single filter are shared, reducing the time and complexity of CNN optimization. Lastly, the result is subjected to an activation function, commonly a hyperbolic tangent, Rectified Linear Units (ReLU), and sigmoid function as shown in Figure 1.

A pooling layer has been frequently used after a convolution layer to obtain a lower resolution representation of the convolution layer activations via sub-sampling. Neighbor pooling units collect input from patches that have been shifted by more than one row or a column so that the representation's dimensions are reduced, and invariance is created to tiny shifts and distortions. Weighted pooling, Max pooling and Mean pooling are the choices for pooling function that calculates statistics on the activations. The fully-connected layer is the next layer in CNN after numerous combinations of convolutional and pooling layers. The fully-connected layer works in the same way as a standard multilayer neural network which can be used in a variety of classification models. Feature extraction is a technique for extracting the optimum features from data to address a particular issue. New features have been produced that capture a dataset's fundamental attributes and represent them in a small space, making learning easier. In the case of a dataset that is too huge with too high processing expenses, this strategy must be used. Using feature extraction, a smaller subset of data is generated that is nonetheless reflective of the input variables. The extraction of this information can assist the model in making better judgments by revealing the links that resulted in output from an input.

Consider a length n and width k of time series. In a multivariate time series, the length denotes the number of time steps, and the width represents the number of variables. For electroencephalography, it might be the number of channels (nodes on a person's head), while for a meteorological time series, it could be factors like temperature, humidity, and pressure. The length of the convolution kernels can be adjusted, but their width always seems to be the same as the time series. Convolution is performed by moving the kernel in one direction from the start to the finish of a time series.

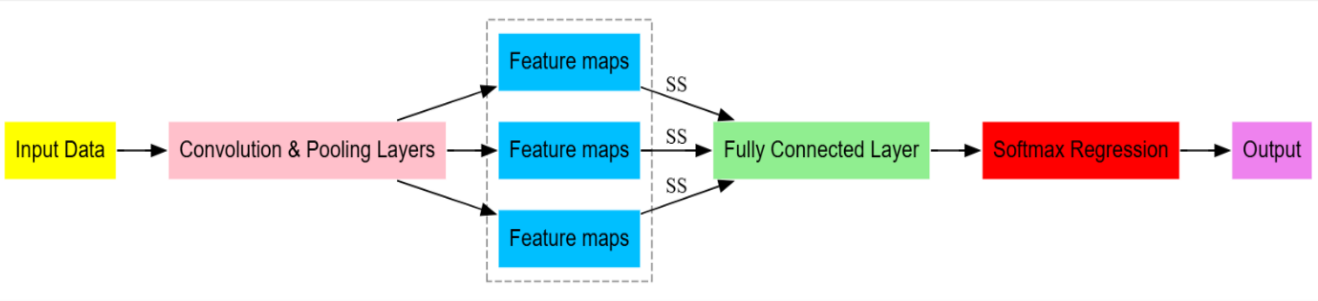

Fig. 1. Block diagram for the proposed work.

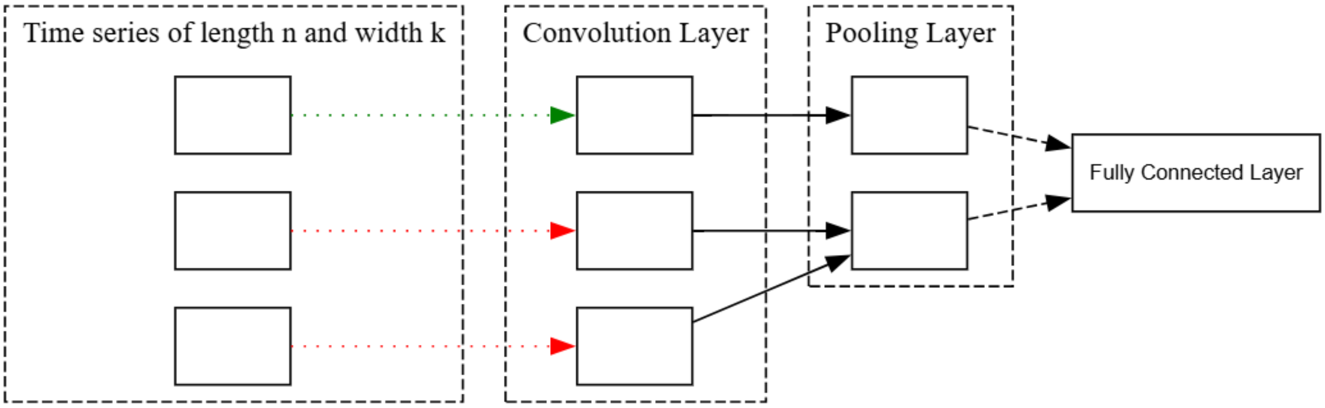

Fig. 2. Proposed work structure analysis.

It will not migrate to the left or right if images are convoluted in two dimensions. The kernel's constituents are multiplied by the time series elements they cover at any given point from Figure 2.

The multiplication results were then combined altogether, and the value was subjected to a nonlinear activation function. The kernel travels forward along the time series to generate the next value, and the resulting value forms an element of a new "filtered" univariate time series. The number of convolution kernels is the same as that of the new "filtered" time series. Various characteristics and features of the initial time series are captured in each of the subsequent filtered series, according to the length of the kernel. The following step is to apply global max-pooling to each of the filtered time series vectors, and the greatest value from each is considered. These values are combined to produce a new vector, and the maximum of these vectors is the resulting feature vector which may be fed into a conventional, fully connected layer as shown in Figure 3.

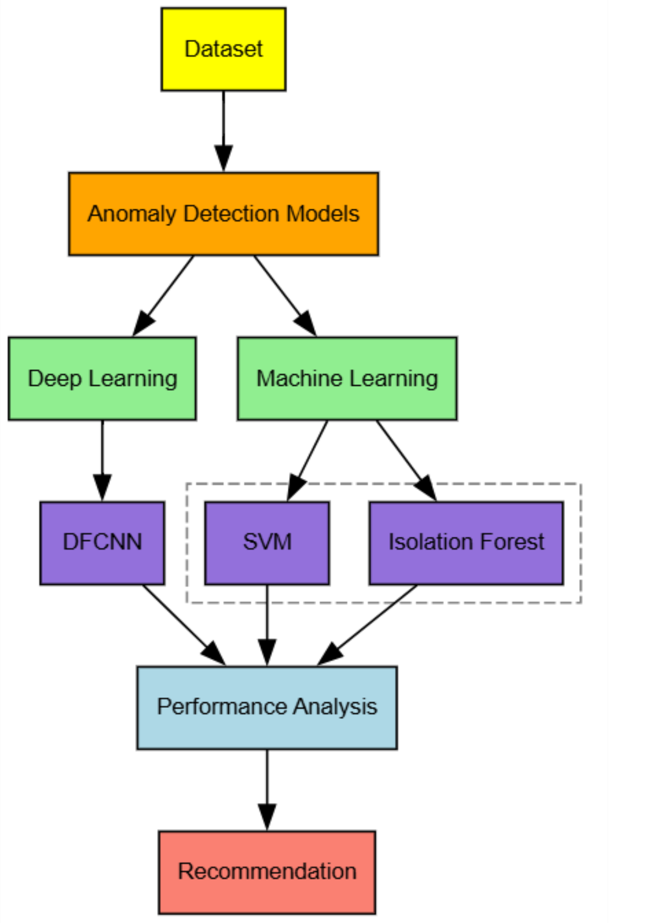

Fig. 3. Development Model Block Diagram of proposed work.

3.1. PRE-PROCESSING

Before transferring the raw data (RW) from the dataset to the input layer, pre-processing is performed. Employing Python tools such as pandas, NumPy, and sklearn, this pre-processing phase removes the null values from the raw input dataset. It also removes noise and duplicated data from RW. The pre-processed output RW_{pre} can be described as the de-noising is performed on the raw data from the dataset first, and null values and duplicate values are removed.

3.2. INPUT LAYER

The pre-processed data RW_{pr} from the previous phase is fed into the input layer. The convolution layer is the most important layer in CNN for feature extraction. This layer extracts the most significant aspects from the previous input layer's data. This layer includes learnable kernels or filters that are employed in the feature extraction process. DFCNN generates the one-dimensional feature maps by performing a convolutional operation on the input data in this layer. Various features of this layer can be retrieved RW^m by using multiple kernels. The kernel detects the specific features in this layer's input feature map at all points. This aids in weight distribution in the feature map. This characteristic of local networking and weight-sharing effectively reduces network complexity and the number of training parameters. For the feature extraction, N convolutional layers $(\mathrm{C_1C_2C_3.CN})$ are used. The work considers five convolutional layers $(\mathrm{N}=5)$ which extract deep features. The number of filters used in these five convolution layers are 128 of size16×1, 64 of size $8\times 1{,}32$ of size $4\times 1{,}16$ of size 4×1, and 8 of size 4×1. Kernel slides are applied to the input in each convolutional layer to generate a feature map. The network structure of the proposed DFCNN is shown in Figure 4. The output of $\mathrm{N^w}$ convolutional layer can be given as,

$$CN_{OP} = \mathrm{ReLU}\big(RW_{\mathrm{pre}} * W_N + b_N\big) \tag{1}$$

Where $CNop$ indicates the output of the $\mathrm{N^{th}}$ convolution layer, RW_{pw} represents the input data, W_N and b_N represents the $\mathrm{N^{th}}$ layer's weight and bias, respectively. The result is subjected to the Rectified Linear Unit (ReLU) activation function following the convolution process.

The neurons are stimulated with the ReLU. This ReLU is vital in neural networks because the input is translated to output in network nodes. It enables the neural network to learn nonlinear dependencies and minimize disappearing gradients with a better learning rate. It also has a faster rate of convergence. In general, linear activation functions are utilized in the output layer of networks for predictions. The ReLU function with input vector x can be written as follows:

$$\mathrm{ReLU}(x) = \max(0, x) \tag{2}$$

3.3. MAX-POOLING LAYERS

In CNN, every convolution layer is followed by a pooling layer. The preceding convolution layer's output will be used as the input for the current pooling layer. $\mathrm{N} = 5$ is used here, and the pooling layers are denoted as $(P_1, P_2, P_3 \ldots P_v)$. Pooling is classified into two types: maximum pooling and average pooling. This max-pooling layer can be used to reduce noise. It may remove noisy activations while also denoising and reducing dimensionality.

In contrast, average pooling reduces dimensionality as part of the noise suppression process. As a result, max-pooling outperforms average pooling. The preceding convolutional layer's output is transmitted to this max-pooling layer, which downsamples the feature map in this max-pooling layer. This layer's features like speed, vertical acceleration, and GPS are extracted from the dataset. The output from the pooling layer can be presented by:

$$P_N = \mathrm{Max}_{N \in S}\, CN_{OP} \tag{3}$$

Where P_N represents the pooled feature map. S denotes the pooling region in the feature map. The DFCNN model is trained using features extracted in the previous layer. The abnormality displayed in CAV is identified in this layer by employing the trained model. When compared to other existing approaches, the proposed DFCNN has a lower training loss and error rate. The output of the fully connected layer is represented as:

$$FCL_{oP} = \mathrm{Predict}(P_N) \tag{4}$$

3.4 OUTPUT LAYER

It is the final layer in the proposed DFCNN, and it indicates whether or not the anomaly is present in the CAV. This layer's final output $\mathrm{OL}_{\mathrm{OP}}$ can be denoted as:

$$OL_{OP} = \begin{cases} 0, \text{ if OP is Normal} \\ 1, \text{ if OP is abnormal} \end{cases} \tag{5}$$

If there is no anomaly in CAV, the output will be 0; otherwise, it will be 1. If an abnormality in the CAV is discovered, rapid action must be performed before the vehicle loses its full control.

3.4.1 Adam Optimization

Adam is the optimization technique employed in this work, and it aids in weight updation utilizing training data. This Adam optimization uses the advantages of Adaptive Gradient (AdaGrad) and Root Mean Square Propagation (RMSProp) techniques. For every parameter, it calculates the individual adaptive learning rate *θ.* Adam optimizer uses the exponentially decaying average of prior gradients m_{i-1}, which is the same as momentum:

$$\begin{aligned} V_i &= \beta_1 V_{i-1} + (1 - \beta_1) g^2 i \\ m_i &= \beta_2 m_{i-1} + (1 - \beta_2) g_i \end{aligned} \tag{6}$$

In the above equations, V represent the variance, and m_i denotes the mean values. The usage of these variables allows Adam's modified rule to be represented as follows:

$$\phi_{i+1} = \theta_i \frac{\mu}{\sqrt{v_l + \varepsilon}} \tag{7}$$

This optimization technique updates weights and selects the best learning rate for accurate prediction.

4. RESULTS AND DISCUSSION

The experiment is carried out on an Intel Core i7 3.5 GHz processor computer with 4 GB of NVIDIA GPU-enabled RAM. The model is implemented in Python utilizing DL frameworks – Keras. CNN models have increasingly been employed in various industries to handle anomaly detection and classification problems. This chapter presented a DFCNN model for identifying anomalies in the CAV. To improve accuracy, the DFCNN model is trained using an instant anomaly category, and the hyperparameters are tuned using a DL approach.

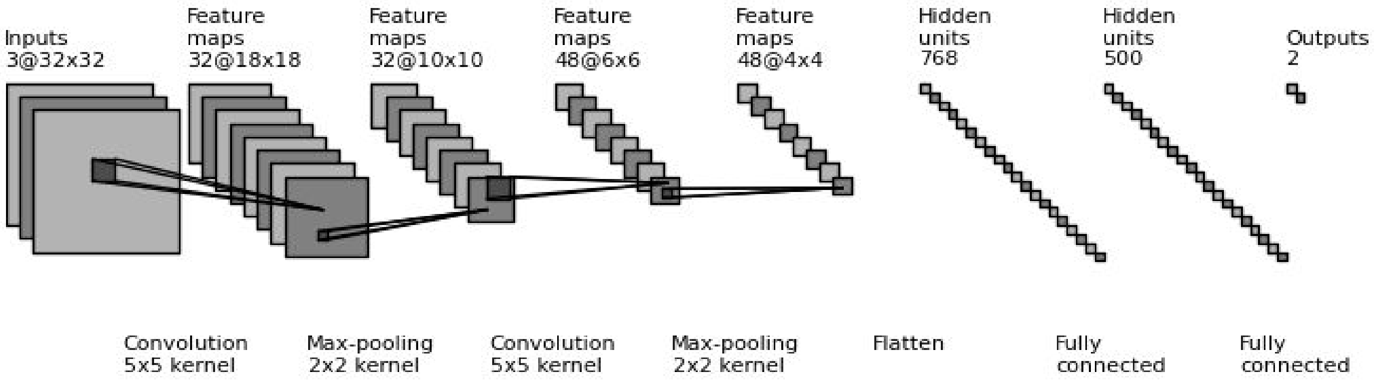

Fig. 4. Network Structure of Proposed DFCNN.

4.1. HYPERPARAMETERS

Hyperparameters are parameters whose values control the learning process and determine the values of model parameters that a learning algorithm ends up learning. Hyperparameter-tuning is essential to find the possible best sets of hyperparameters to build the model from a specific dataset. The process of training a model involves choosing the optimal hyperparameters that the learning algorithm will use to learn the optimal parameters that correctly map the input features (independent variables) to the labels or targets (dependent variable) such that some form of intelligence can be achieved. The hyperparameters of the proposed DFCNN are listed in Table 1.

Table 1. Hyperparameters of the Proposed DFCNN.

Hyperparameter	Value
Learning rate	0.001
Batch size	100
Epochs	100
Filters	128
Optimizer	Adam
Activation function	ReLU

4.2. CONFUSION MATRIX

A confusion matrix is used to evaluate the efficiency of the proposed method on the test data. The rows of the confusion matrix contain information regarding the true class, while the columns contain information regarding the predicted class from Figure 5**.**

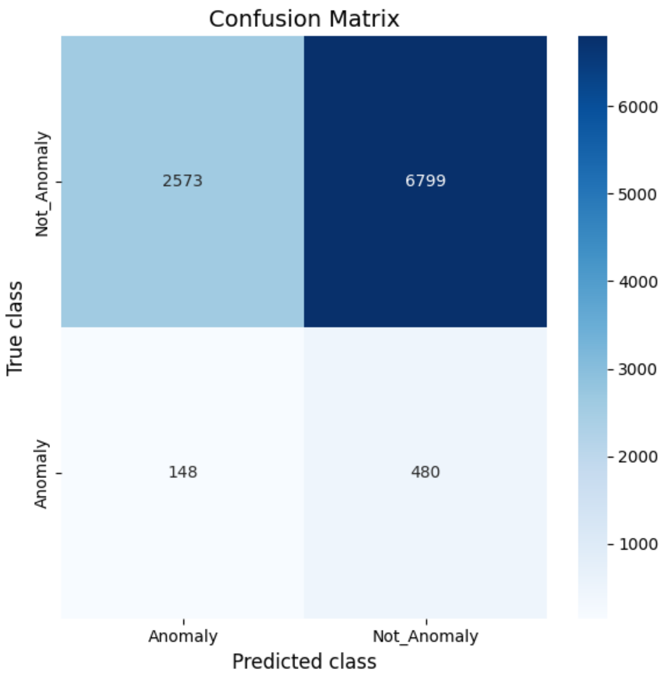

Fig. 5. Confusion Matrix for the Proposed DFCNN.

There are four outputs in this matrix: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). When the output is positive, TP denotes that it belongs to the positive class category. When the output is negative, TN signifies that it belongs to the negative class category. When the outcome is negative, FP signifies that it belongs to the positive class category. When FN represents a negative outcome, it relates to the negative class category.

The number of FPs and FNs vary each class due to class differences and the number of data sets. The confusion matrix can be used to determine the TP, TN, FP, and FN values at each position on the test dataset. We extracted 12,000 records from the SPMD dataset. 1250 data points are used as testing data, 1250 data points as validation data, and 10000 data points as training data. TP value=2573 is the confusion matrix, indicating that the model accurately predicts the anomaly in CAV as an anomaly. TN = 480, indicating that the model accurately predicts no anomaly in CAV. Likewise, the FN= 6799 indicates that the model properly predicts no anomaly in CAV. FP=148, indicating that the model inaccurately predicts an anomaly in CAV. These TP, TN, FP, and FN values are calculated using the confusion matrix.

4.3 ROC CURVE

The ROC curve appears to be an essential metric for problem classification and identification from Figure 6. This ROC is a probability curve that is used to plot the True Positive Rate (TPR) vs the False Positive Rate (FPR) at several threshold values to separate the signal from the noise. TPR, also known as sensitivity, shows how well the negative class is predicted. The FPR or specificity indicates how much of the model's negative class incorrectly predicts.

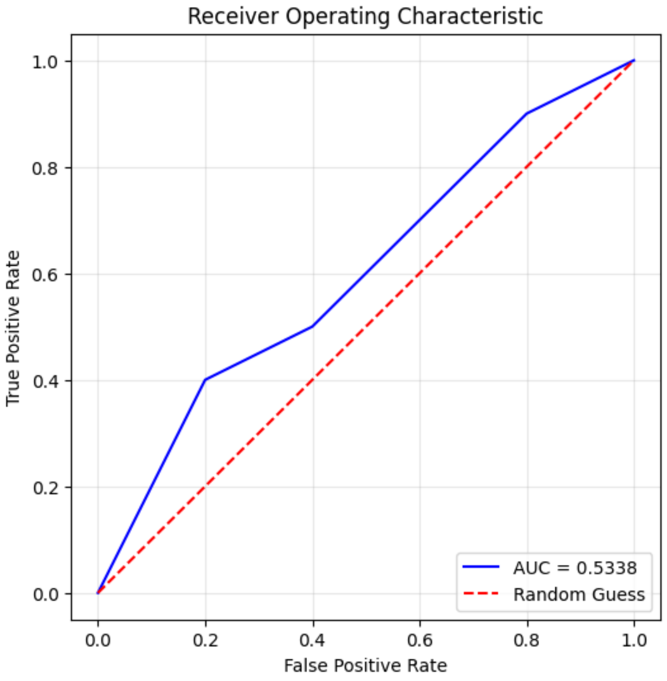

Fig. 6. ROC Curve for the Proposed DFCN.

The Area Under the Curve (AUC) is a measure of a model's ability to distinguish between groups and is used to summarize the ROC curve. The AUC of the proposed work is 0.5338, and this high value suggests that the model's output in dividing between positive and negative groups is higher. In Figure 7, the ROC curve is closer to the upper left corner, indicating that the proposed work detects the anomaly in CAV more correctly than the existing techniques.

According to Table 2, the proposed DFCNN has a high true positive rate compared to other existing methods such as isolated forest and SVM. It is evident that the proposed method successfully identifies many anomalies while producing a small number of false positives. Because the model is given robust training with a large amount of data, the proposed DFCNN detects anomalies correctly. Before training, the null values are removed during the pre-processing stage.

Table 2. ROC Curve for the Proposed DFCNN and Existing Models.

False Positive Rate	True Positive Rate		
	Proposed DFCNN	Isolation Forest	SVM
0.0	0.0	0.0	0.0
0.2	0.18	0.16	0.14
0.4	0.42	0.39	0.37
0.6	0.64	0.62	0.59
0.78	**0.9**	**0.87**	**0.85**

4.4. ACCURACY, PRECISION, AND RECALL

The precision-recall curve is generated using the confusion matrix from the test dataset. The DFCNN has a high AUC score, which implies that the model is producing accurate (high accuracy) and mostly positive outcomes (high Recall). Precision is defined as the proportion of accurately labeled positive samples to the total number of positive samples classified (either correctly or incorrectly). The accuracy metric evaluates the model's ability to correctly interpret a result as positive. Precision ranges from 0 to 1. A false positive in anomaly detection indicates that a CAV is not under attack (actual negative) and has been recognized as being under attack (predicted anomaly). When the precision for the anomaly detection model is not high, the CAV loses complete control over the attacker. The Recall is used to calculate the number of correct positive predictions by dividing the number of true positive findings by the total number of samples. Figure 7 and Figure 8 compare the proposed DFCNN's validation accuracy, precision, and recall values to those of existing models such as IF and SVM. The greater the Recall, the more accurate the anomaly detection. For extracting significant features, the five convolutional and max-pooling layers are used. This results in more accurate detection accuracy due to more accurate training. The proposed DFCNN obtains a precision value of 97.1 percent, which is greater than existing models, showing that the proposed method performs better.

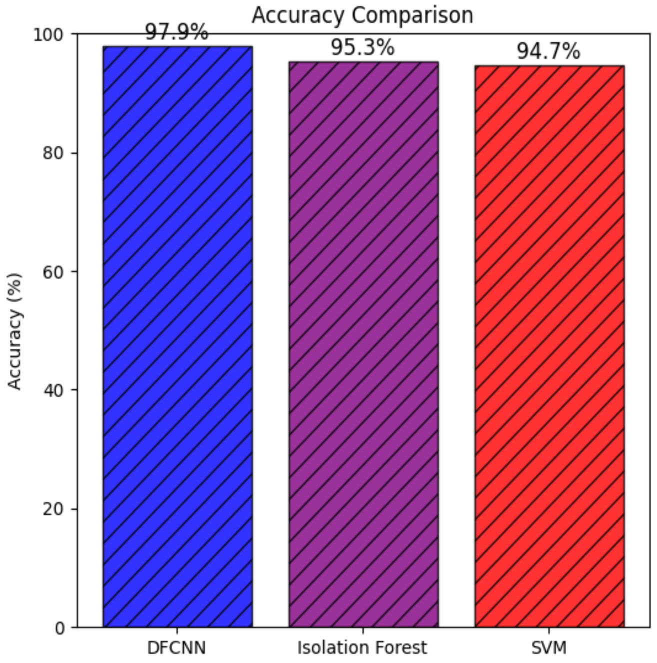

Fig. 7. Accuracy Comparison of DFCNN with IF and SVM.

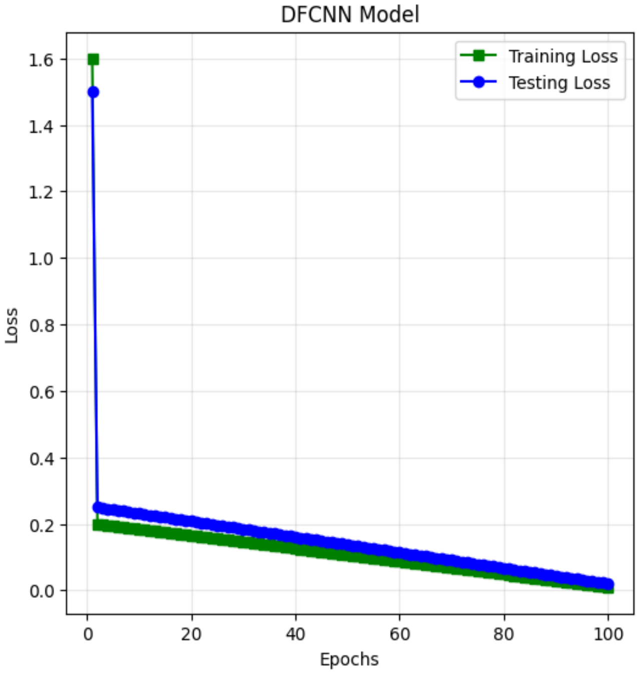

Fig. 8. Comparison of Precision and Recall of DFCNN with IF and SVM.

Furthermore, the new DFCNN obtains a recall value of 98.7 percent, which is greater than previous models, demonstrating the improved performance of the proposed technique.

4.5 TRAINING AND TEST LOSS

The loss function is one of the essential components of neural networks, and it is a prediction error of the model. Training data is used to train the model. Later, during the detection of an anomaly in CAV, test data is used. Model loss is calculated at both of these stages. The model loss on the training and test dataset for DFCNN is illustrated in Figure 9.

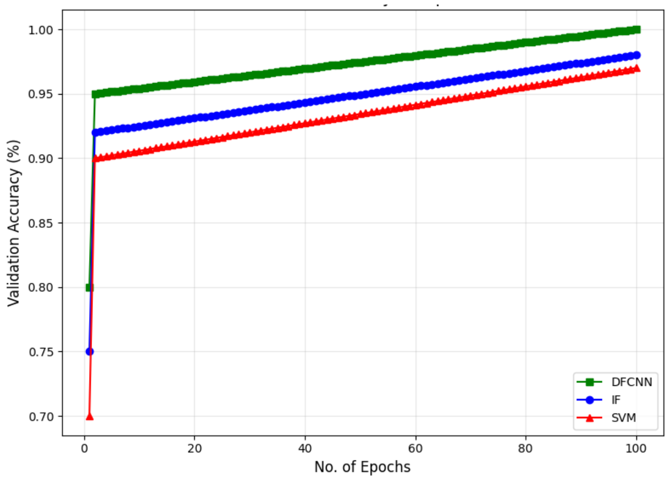

Fig. 9. DFCNN Training and Testing Loss Comparison.

This shows the model loss for epochs = 100. By modifying the weight vector values and utilizing the Adam optimization method, the value of loss function value is reduced with regard to the model's parameters. Figure 10 indicate the model loss of isolation forest and SVM. The training loss and training loss of these two methods are higher than the proposed DFCNN. Due to proper training with a large dataset, the proposed DFCNN has low test loss compared to existing models.

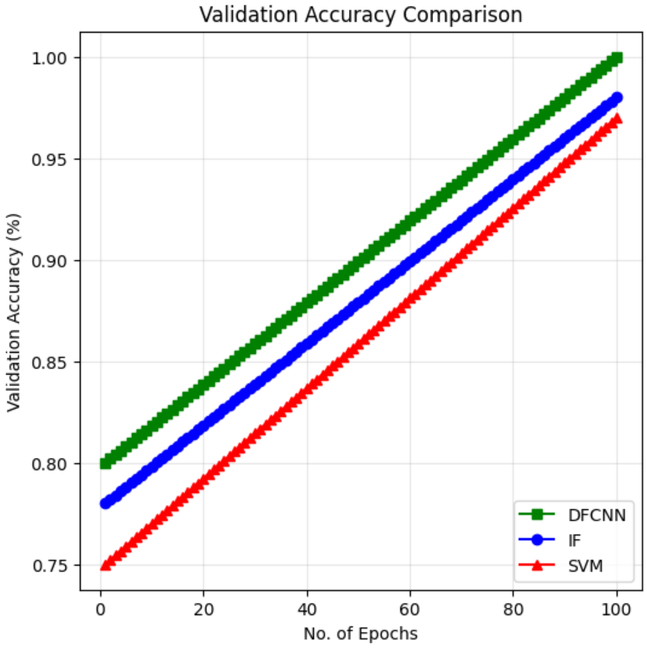

Fig. 10. IF Training and Testing Loss Comparison.

4.6 TRAINING AND VALIDATION ACCURACY

The training accuracy and the validation accuracy of the proposed DFCNN are compared with IF and SVM models. When compared to IF and SVM, the training accuracy of DFCNN is higher, which is shown in Figure 11

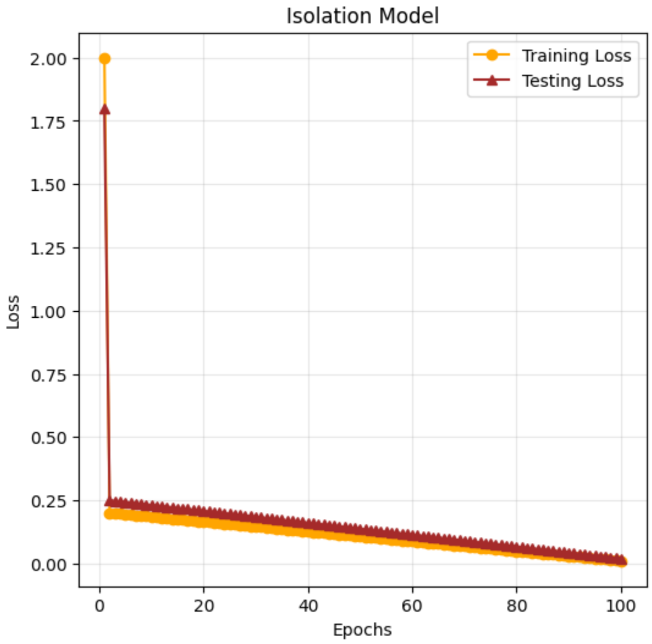

Fig. 11. SVM Training and Testing Loss Comparison

Figure 12 shows that the proposed DFCNN has higher validation accuracy when compared to IF and SVM models. This shows the model accuracy for 100 epochs. The proposed DFCNN achieves a highest validation accuracy of 97.9% over other models which is 2.6% higher than IF and 3.2% higher than other models.

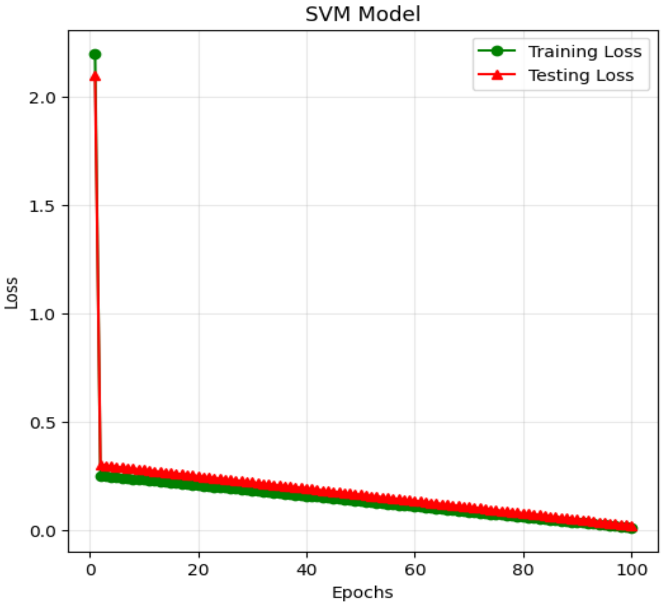

Fig. 12. Comparison of Training Accuracy.

5. CONCLUSION

Connected and Autonomous Vehicles (CAVs) signify a pivotal advancement in the automobile industry, bridging the gap between automation, connectivity, and enhanced road safety. These vehicles leverage sophisticated driver assistance systems and cutting-edge automated driving technologies to mitigate human errors, reduce collisions, and optimize traffic flow through real-time navigation and intelligent decision-making. The potential of CAVs to revolutionize transportation is immense, promising safer roads, improved mobility, and greater convenience for users. However, the widespread adoption of CAVs is accompanied by critical challenges, particularly in the realms of cybersecurity and system reliability. The interconnected nature of these vehicles, coupled with their reliance on vast networks of sensors and communication systems, makes them susceptible to hacking and cyber threats. Such vulnerabilities could compromise not only individual vehicles but also the broader transportation ecosystem. Therefore, addressing these concerns requires the integration of robust security protocols, advanced encryption methods, and continuous monitoring systems to safeguard data and prevent unauthorized access. Additionally, CAVs must overcome obstacles related to infrastructure readiness, regulatory frameworks, and societal acceptance. Investments in smart road networks, government policies to guide autonomous vehicle deployment, and public awareness campaigns are essential for ensuring their smooth integration into daily life. As CAV technologies mature, their transformative impact on transportation will become increasingly apparent, offering safer, more efficient, and environmentally friendly mobility solutions. By embracing adaptive AI architectures and prioritizing security innovations, CAVs can pave the way for a future where transportation is both intelligent and resilient.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interests.

REFERENCES

[1] Hsu, F.H., Wang, C.S., Hsu, Y.L., Cheng, Y.P. and Hsneh, Y.H., **2017.** A client-side detection mechanism for evil twins. *Computers & Electrical Engineering*, *59*, pp.76-85.

[2] Rane, J., Mallick, S.K., Kaya, O. and Rane, N.L., **2024.** Scalable and adaptive deep learning algorithms for large-scale machine learning systems. Future Research Opportunities for Artificial Intelligence in Industry 4.0 and, 5, pp.2-40.

[3] Baratchi, M., Wang, C., Limmer, S., van Rijn, J.N., Hoos, H., Bäck, T. and Olhofer, M., **2024.** Automated machine learning: past, present and future. *Artificial Intelligence Review*, *57*(5), pp.1-88.

[4] Wang, Y., Hou, M., Plataniotis, K.N., Kwong, S., Leung, H., Tunstel, E., Rudas, I.J. and Trajkovic, L., **2020.** Towards a theoretical framework of autonomous systems underpinned by intelligence and systems sciences. *IEEE/CAA Journal of Automatica Sinica*, *8*(1), pp.52-63.

[5] Fuchs, A., Passarella, A. and Conti, M., **2023.** Modeling, replicating, and predicting human behavior: a survey. *ACM Transactions on Autonomous and Adaptive Systems*, *18*(2), pp.1-47.

[6] Li, Y., Wang, H., Dang, L.M., Nguyen, T.N., Han, D., Lee, A., Jang, I. and Moon, H., **2020.** A deep learning-based hybrid framework for object detection and recognition in autonomous driving. *IEEE Access*, *8*, pp.194228-194239.

[7] Shaheen, K., Hanif, M.A., Hasan, O. and Shafique, M., **2022.** Continual learning for real-world autonomous systems: Algorithms, challenges and frameworks. Journal of Intelligent & Robotic Systems, 105(1), p.9.

[8] Akhunzada, A., Al-Shamayleh, A.S., Zeadally, S., Almogren, A. and Abu-Shareha, A.A., **2024.** Design and performance of an AI-enabled threat intelligence framework for IoT-enabled autonomous vehicles. *Computers and Electrical Engineering,* 119, p.109609.

[9] Santos, M., **2023.** Hybrid Control Architectures for Autonomous Systems-Analyzing hybrid control architectures combining classical and learning-based approaches for autonomous systems. *Journal of Computational Intelligence and Robotics*, 3(1), pp.1-14.

[10] Thakur, A. and Mishra, S.K., **2024.** An in-depth evaluation of deep learning-enabled adaptive approaches for detecting obstacles using sensor-fused data in autonomous vehicles. *Engineering Applications of Artificial Intelligence*, 133, p.108550.

[11] Carlucho, I., De Paula, M., Wang, S., Petillot, Y. and Acosta, G.G., **2018.** Adaptive low-level control of autonomous underwater vehicles using deep reinforcement learning. *Robotics and Autonomous Systems,* 107, pp.71-86.

[12] Tanimu, J.A. and Abada, W., **2024.** Addressing Cybersecurity Challenges in Robotics: A Comprehensive Overview. *Cyber Security and Applications*, p.100074.

[13] Hao, C. and Chen, D., **2021**, June. Software/hardware co-design for multi-modal multi-task learning in autonomous systems. In 2021 *IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems* (AICAS) (pp. 1-5). IEEE.

[14] Sifakis, J. and Harel, D., **2023**. Trustworthy autonomous system development. *ACM Transactions on Embedded Computing Systems*, 22(3), pp.1-24.

[15] Pramudito, D.K., **2024.** Enhancing Real-Time Object Detection in Autonomous Systems Using Deep Learning and Computer Vision Techniques. *The Journal of Academic Science*, 1(6), pp.788-804.

[16] He, H., Gray, J., Cangelosi, A., Meng, Q., McGinnity,

T.M. and Mehnen, J., **2021.** The challenges and opportunities of human-centered AI for trustworthy robots and autonomous systems. *IEEE Transactions on Cognitive and Developmental Systems*, 14(4), pp.1398-1412.

[17] Panella, I., Fragonara, L.Z. and Tsourdos, A., **2021.** A deep learning cognitive architecture: Towards a unified theory of cognition. In Intelligent Systems and Applications: Proceedings of the 2020 Intelligent Systems Conference (IntelliSys) Volume 1 (pp. 566-582). Springer International Publishing.

[18] Zhang, T., Li, Q., Zhang, C.S., Liang, H.W., Li, P., Wang, T.M., Li, S., Zhu, Y.L. and Wu, C., **2017.** Current trends in the development of intelligent unmanned autonomous systems. *Frontiers of Information Technology & Electronic Engineering*, 18, pp.68-85.

[19] Kuhner, D., Fiederer, L.D.J., Aldinger, J., Burget, F., Völker, M., Schirrmeister, R.T., Do, C., Boedecker, J., Nebel, B., Ball, T. and Burgard, W., **2019.** A service assistant combining autonomous robotics, flexible goal formulation, and deep-learning-based brain–computer interfacing. *Robotics and Autonomous Systems*, 116, pp.98-113.

[20] Alahmed, S., Alasad, Q., Hammood, M.M., Yuan, J.S. and Alawad, M., **2022.** Mitigation of black-box attacks on intrusion detection systems-based ml. *Computers*, *11*(7), p.115.