American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-13, Issue-10, pp: 63-75
www.ajer.org
Research Paper Open Access

Training a Gear Shifting in Automated Transmissions Using a Perceptron-Based Neural Network: A Low Computational Cost Approach

Vinicius Teodoro Elzinga[1], *Sérgio Junichi Idehara[1]
[1]*Federal University of Santa Catarina, Joinville Campus*
**Corresponding Author*

ABSTRACT : *This study focuses on the development and implementation of a Perceptron-based Artificial Neural Network (ANN) for optimizing gear shifting in automated transmissions. The primary objective was to characterize the vehicle dynamics of a manual powertrain and explore the feasibility of employing artificial intelligence to enhance decision-making in gear shifting, particularly in retrofitting vehicles to automated transmissions. The ANN was trained and tested using experimental data, demonstrating its capability to learn and accurately replicate driving patterns. The results showed that the implemented Perceptron network algorithm achieved high accuracy (over 99%) in gear selection across various dynamic driving conditions, highlighting the robustness of the neural network. A significant advantage of this approach is its low computational cost. The simplicity of the Perceptron model significantly reduces the computational resources required, making it particularly suitable for real-time applications where processing power and response time are critical. Compared to more complex neural network architectures, the Perceptron-based method offers an optimal balance between precision and computational efficiency, providing an effective solution for integration into automotive control systems. This study not only advances vehicle performance by accurately modeling driver behavior but also presents a cost-effective and scalable approach for implementing advanced vehicle control strategies. Future research could further enhance vehicle performance by integrating this method with other automotive systems, paving the way for more intelligent, efficient, and adaptive vehicles.*

KEYWORDS *gear shifting, automated transmission, artificial neural network, perceptron model.*

Date of Submission: 03-10-2024 Date of acceptance: 16-10-2024

I. INTRODUCTION

Manual transmissions (gearboxes) require the driver to manually operate the clutch and gear selector to efficiently transfer engine torque. However, unlike automatic transmissions or continuously variable transmissions (CVT), which suffer from torque energy loss due to the slipping nature of torque converters or variable-diameter pulleys, manual transmissions provide a direct mechanical link between the engine and wheels, eliminating this inefficiency [1]. Manual transmissions also have relatively lower production and maintenance costs and are almost half the weight of an automatic transmission. Nevertheless, a manual gearbox requires skillful operation to achieve optimal vehicle performance and can be more exhausting, especially in heavy traffic, which can be challenging for drivers.In this context, an automated manual transmission combines the efficiency of manual transmissions with the convenience of automatic transmissions. Automated transmissions use actuators (hydraulic or electric) to automate these processes, eliminating the need for skilled clutch operation while maintaining the high power transmission efficiency inherent in manual systems, which can be around 96% [2]. This type of gearbox also avoids the complexity and potential maintenance issues associated with hydraulic systems, such as oil leakage and viscosity changes with temperature. However, converting a manual gearbox to an automated one requires an algorithm to predict the optimal timing for gear shifts. In manual transmission systems, the driver manually selects the appropriate gear based on driving conditions. Automated transmission systems, on the other hand, utilize algorithms to determine the optimal timing for gear shifts, aiming to enhance driving behavior. Despite their advantages, conventional automated systems often lack the sophistication needed to make real-time adjustments based on driver needs and preferences.

With the advent of advanced computational tools, traditional approaches are increasingly being supplemented or replaced by machine learning algorithms. These methods enable the analysis of complex datasets and the development of accurate models for various aspects of vehicle operation. A growing body of research highlights the application of machine learning in automotive systems, particularly in fuel consumption estimation and gear shifting optimization. For instance, Abediasl et al. [3] demonstrated the use of machine learning algorithms to estimate real-time fuel consumption using on-board diagnostics (OBD) data, achieving high accuracy and practical application through models such as Artificial Neural Networks (ANN) and Random Forests (RF) in capturing real-world driving conditions.The integration of machine learning techniques into automotive systems has also opened other possibilities for optimizing vehicle performance, particularly in areas such as regenerative braking. For example, Hwang et al. [4] explored AI-based regenerative braking control strategies to enhance driving comfort in autonomous vehicles, demonstrating how a backpropagation method can fine-tune vehicle responses to dynamic conditions by estimating braking force limits. Lv et al. [5] introduced a hybrid learning approach that combines unsupervised and supervised learning methods to classify and quantitatively infer driver braking intensity in electrified vehicles. By utilizing a Gaussian Mixture Model (GMM) for clustering and a Random Forest (RF) model for classification, along with an Artificial Neural Network (ANN) for continuous brake pressure estimation, this methodology effectively recognized and predicted braking behaviors under various driving conditions. This approach improved the accuracy of braking intensity predictions and demonstrated the potential for sensorless braking systems, significantly reducing system costs while enhancing safety.

This approach aligns with a broader trend in automotive engineering, where multi-objective optimization is increasingly being employed to enhance powertrain components, as demonstrated by Kwon et al. [6]. Their work on multi-speed transmissions for electric vehicles underscores the importance of integrating efficiency analysis across various powertrain components, including inverters, motors, and transmissions, to achieve optimal performance. Shukla and Sharma [7] proposed a neural network-based variable DC-link voltage control scheme for induction motors in EV applications, showing significant improvements in torque and speed performance across various operating conditions. Their study highlighted the effectiveness of ANNs in minimizing inverter switching losses and enhancing overall efficiency by dynamically adjusting the DC-link voltage. The application of ANN-based control strategies not only improved motor performance under different load conditions but also reduced torque ripples. Kwon et al. [8] further demonstrated the efficacy of integrating artificial neural networks (ANNs) into the design and optimization processes of EV powertrains. By utilizing an ANN-based optimization process, they effectively reduced computational efforts while enhancing the energy efficiency and dynamic performance of multi-motor and multi-speed powertrain configurations. This research underscores the potential of ANN-driven methodologies to address the complexities of optimizing vehicle systems, particularly in balancing the trade-offs between efficiency and performance.

Yang et al. [9] demonstrated the application of an ANN model to predict multiple engine parameters, including power, emissions, and combustion phasing, using input variables such as engine speed, intake pressure, and spark timing. Their study confirmed that a single ANN model could effectively integrate and predict various engine metrics, achieving high accuracy with R^2 values exceeding 0.98. This integration of multiple engine parameters into one predictive model underscores the potential of ANNs to enhance engine design processes and optimize powertrain control strategies in hybrid vehicles. Similarly, Kanchev et al. [10] utilized a NARMA (Nonlinear AutoRegressive Moving Average) neural network to control a hybrid electric vehicle's traction system, comprising a battery, ultracapacitor, and Brushless DC motor. Their study highlighted the advantages of neural network-based control strategies in managing energy flows, particularly in optimizing regenerative braking and improving overall vehicle dynamics. The research demonstrated that these advanced control methods could outperform traditional PI controllers, offering better robustness, shorter transient response times, and reduced overshoot.The use of artificial neural networks (ANNs) in predicting engine performance and emissions has become increasingly prevalent in recent years, driven by the need to optimize internal combustion engines for both efficiency and environmental compliance. For example, Park et al. [11] employed an ANN to predict the composition of reformed gas in diesel engines equipped with an Exhaust Gas Recirculation (EGR) system. Their results demonstrated that the ANN could accurately predict hydrogen and carbon monoxide concentrations, leading to optimized reforming processes that significantly reduced NOx and particulate matter (PM) emissions. This approach to controlling emissions underscores the broader applicability of ANNs in automotive engineering, where control and optimization are important.Similarly, d'Apolito and Hong [12] used ANNs to estimate fuel consumption for forklift trucks under various driving cycles, utilizing simulations performed with AVL Cruise software. The ANN model, trained on these simulation results, effectively predicted fuel consumption across a wide range of driving scenarios, demonstrating the power of machine learning in reducing the need for extensive physical prototyping and testing. In another study, Seo et al. [13] developed an integrated ANN and vehicle dynamics model to predict instantaneous emissions of carbon dioxide (CO_2), nitrogen oxides (NOx), and total hydrocarbons (THC) from diesel light-duty vehicles. By leveraging real-world driving data and a multi-layer feed-forward ANN, the study achieved high prediction accuracy for these emissions, emphasizing the importance of selecting the optimal combination of input variables such as engine speed, engine torque, and vehicle velocity.

The synthesis and optimization of gearboxes have long been critical areas of research in automotive engineering. Saeed [14] demonstrated the effectiveness of a backpropagation neural network implemented on a Field Programmable Gate Array (FPGA) for controlling the speed ratios in an automatic transmission gearbox. By using a neural network with eight input neurons, five hidden neurons, and five output neurons, the system accurately selected the appropriate gear ratio based on the vehicle's real-time speed data. The study highlighted the advantages of MATLAB's Levenberg-Marquardt training function and linear activation functions in achieving high accuracy and fast processing times. As noted by Petrescu et al. [15], despite the emergence of advanced gearbox types such as automatic, semi-automatic, and continuously variable transmissions, the classic manual gearbox remains widely used in many vehicles. Optimizing these systems is crucial not only for improving mechanical performance but also for enhancing overall vehicle efficiency. For instance, Brazil had an estimated car fleet of about 62 million vehicles in 2023 [16], with the majority of them based on traditional manual transmission systems.

The dynamic synthesis of a manual gearbox involves accurately determining its mechanical performance under various operating conditions, thereby achieving optimal synthesis regardless of the gearbox's status. This study proposes an approach to gear shifting in automated transmissions using a Perceptron-based ANN with low computational cost, trained on real-time vehicular data acquired through OBD systems. By optimizing the gear selection process—one of the possible options in shifting strategies based on driver experience—the model aims to improve the overall driving experience and vehicle performance, while reducing the physical demands on the driver and enhancing product acceptability. The methodology and results presented in this paper align with current trends in automotive innovation, where the integration of machine learning into vehicle control systems is becoming increasingly prevalent.

II. ARTIFICIAL NEURAL NETWORK

The architecture of a neural network can vary significantly depending on the computational requirements needed to solve a particular problem, with its nodes organized into distinct layers. The simplest form of a neural network is the perceptron, which consists of an input layer that processes stored information into a single output node [17], as shown in Fig. 1. The Perceptron neural network used in this work features an input layer, a hidden layer, and an output layer, as illustrated in Fig. 1. The input and output layers have the same number of neurons as the respective input and output variables. The number of neurons in the hidden layer is determined by summing the neurons in the input and output layers. The weights between the input and hidden layers ($W^1_{i,j}$ and b_{h_i}) and between the hidden and output layers ($W^2_{j,k}$ and b_{y_k}) are initially assigned random values, typically within a uniform distribution range of -0.05 to 0.05, as the starting point for training.

The calculation of the output vector (y_k) from the output of the hidden layer (h_j) is performed using a sigmoid function σ:

$$y_k = \sigma\{z_{y_k}\} = \sigma\left\{\sum_j \left(W^2_{j,k}.h_j\right) + b_y(k)\right\} \tag{1}$$

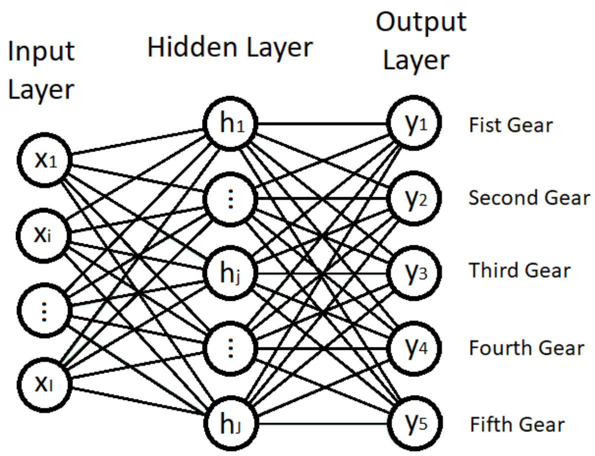

Fig. 1- Illustration of the artificial neural network architecture

The calculation of the hidden layer (h_j) from the input data is given by:

$$h_j = \sigma\left\{z_{h_j}\right\} = \sigma\left\{\sum_i \left(W_{i,j}^1 . x_i\right) + b_h(j)\right\} \tag{2}$$

The weights $W_{i,j}^1$ and $W_{j,k}^2$ are the coefficients that multiply the values from the previous layer, summed together and added to independent terms $b_h$and b_y, respectively, to determine the inputs for the next layer. The resulting value at each neuron is then passed through the sigmoid (logistic) activation function, which is mathematically defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

The calculation of the coefficients is performed using the Back Propagation method, which involves determining the gradient of the error function generated during each iteration of parameter adjustment. This approach maintains low computational costs, even when a high number of patterns are employed in the training phase.

BACKPROPAGATION ALGORITHM

The calculation of weight coefficients begins with the desired output data and works backward to the input data. This iterative procedure starts with random initial values and determines a set of coefficients that minimize the quadratic error between the actual data and the estimated results from the ANN.The first part of the algorithm determines the final coefficients $W_{j,k}^2$, between the hidden and output layers. The error between the actual value t_k (selected gear) and the estimated gear $y_k$is given by:

$$E_{y_k} = \sum\nolimits_k \frac{1}{2}(t_k - y_k)^2 \tag{4}$$

Then, the gradient of the error is a vector of the partial derivatives of the weights $W_{j,k}^2$.

$$\frac{\partial E_{y_k}}{\partial W_{j,k}^2} = \frac{\partial E_{y_k}}{\partial y_k} . \frac{\partial y_k}{\partial z_{y_k}} . \frac{\partial z_{y_k}}{\partial W_{j,k}^2} \tag{5}$$

Where,

$$\frac{\partial E_{y_k}}{\partial y_k} = y_k - t_k \tag{6}$$

$$\frac{\partial y_k}{\partial z_{y_k}} = y_k(1 - y_k) \tag{7}$$

$$\frac{\partial z_{y_k}}{\partial W_{j,k}^2} = h_j \tag{8}$$

The value of h_j, from the hidden layer output, is computed using the previous coefficients values. For updating the weight coefficients based on the learning rate η and momentum α, the equation is:

$$W_{j,k}^2(p) = W_{j,k}^2(p-1) - \eta . \frac{\partial E_{y_k}}{\partial W_{j,k}^2} - \alpha . \left[W_{j,k}^2(p-1) - W_{j,k}^2(p-2)\right] \tag{9}$$

Where p represents the iteration number.

The second part of the algorithm involves determining the weight coefficients between the input and hidden layers, $W_{i,j}^1$. Similar to the first part, the error gradient is represented as:

$$\frac{\partial E_{h_j}}{\partial W_{i,j}^1} = \frac{\partial E_{h_j}}{\partial h_j} . \frac{\partial h_j}{\partial z_{h_j}} . \frac{\partial z_{h_j}}{\partial W_{i,j}^1} \tag{10}$$

Which can be rewritten as:

$$\frac{\partial E_{h_j}}{\partial W_{i,j}^1} = \left\{\sum\nolimits_k (y_k - t_k)[y_k(1 - y_k)] . W_{j,k}^2\right\} . \left[h_j\left(1 - h_j\right)\right] . x_i \tag{11}$$

And the updated coefficients between the input and hidden layers are:

$$W_{i,j}^{1}(p) = W_{i,j}^{1}(p-1) - \eta . \frac{\partial E_{h_j}}{\partial W_{i,j}^{1}} - \alpha . \left[W_{i,j}^{1}(p-1) - W_{i,j}^{1}(p-2)\right] \quad (12)$$

The algorithm iteratively calculates the sets of coefficients $W_{j,k}^{2}$ and $W_{i,j}^{1}$ until the square error converges.

The learning rate η and momentum α are parameters that vary between zero and one and are used to control the precision and speed of iterations in the network. The learning rate influences the extent to which each neuron's weight is adjusted during iterations. A lower learning rate reduces the weight changes between iterations, enhancing the network's stability but at the cost of slower convergence [18]. Conversely, increasing the learning rate speeds up the calculation process but may cause the network to become unstable. The momentum term helps mitigate high variations between iterations, enabling the use of a higher learning rate. Momentum smooths out updates by maintaining the direction of the weight changes from previous steps, reducing the likelihood of getting stuck in local minima and making the learning process faster and more stable, particularly when dealing with complex error surfaces. In this work, the learning rate was set to $\eta = 0.3$ and the momentum to $\alpha = 0.9$.

NUMBER OF MATHEMATICAL OPERATIONS

The layout of the Perceptron ANN and the backpropagation calculation result in relatively lower computational effort. By counting the operations in the algorithm, the key computations include matrix-vector multiplications, bias additions, activation function operations, subtractions, element-wise squaring, sums, and multiplications in error calculation, as well as multiplications and subtractions during weight updates. Additionally, there are loops over the number of data samples and iterations, further influencing the computational demand.

Considering I as the number of input neurons, J as the number of hidden neurons, K as the number of output neurons, N_d as the length of the training data, and N_i as the number of iterations, the estimated total number of operations is:

$$O(N_i \times N_d \times I \times J \times K) \cong N_i \times N_d \times [(I \times J) + (3 \times J) + (J \times K) + (3 \times K) + (6 \times J \times K) + (I \times J \times K)] \quad (13)$$

The complexity grows linearly with the number of iterations, data points, and the size of the neural network (in terms of input, hidden, and output neurons). This reflects the combined cost of forward and backward propagation during training. Other types of ANN require more computational effort, such as a Convolutional Neural Network (CNN), which has complexity $O(I \times L \times M^2)$, where L is the filter size and M is the feature map size. The M^2 term in convolutional operations results in quadratic growth.

III. METHODOLOGY

The ANN training process involves using vehicle parameters to specify the ideal gear. These parameters include engine rotational speed, vehicle speed, and the percentage of torque used, relative to the maximum engine torque. All data are obtained in real-time conditions via the OBD-II (On-Board Diagnostics) connector from the Freematics One+ (Fig. 2). The data acquisition sample rate was 0.6 Hz.

Fig. 2- Photo of the data logger used in the vehicle's OBD-II port

Before the ANN training process, the vehicle speed, engine rotation, and torque data were normalized using their maximum values across the entire dataset. Initial random coefficients were generated in Matlab software to train the neural network and determine the coefficients based on experimental vehicle tests. Fig. 1 illustrates the architecture of the ANN, with the input layer, J nodes in the hidden layer, and five nodes in the output layer. The number of nodes in the hidden layer was chosen as the sum of the input and output nodes. The output layer was configured as a binary response, where the selected gear has a value of one, and the others have a value of zero. For example, the first gear is represented in the output layer by [1,0,0,0,0], the second gear by [0,1,0,0,0], and so on for the other three gears. Initially, the gear selection was represented by the total transmission ratio value for training the neural network. However, the binary format yielded better results with the experimental data.

For this work, the ANN coefficients have a size of 3x8 for W_{ij}^1 and 8x5 for W_{jk}^2. The input vector consists of 3 nodes, and the output vector consists of 5 nodes.

IV. VEHICULAR TESTING

The test vehicle is a passenger car with a manual gearbox, a 1.6-liter flex-fuel engine producing 98hp of power and 142Nm of torque. The unladen vehicle weight is 1,063kg. During the tests, the vehicle weight varied between 1,120kg and 1,480kg. The total driveline ratio is:

Table 1- Total driveline ratio for each gear.

Gear	1st	2nd	3rd	4th	5th
Ratio	15.18	8.34	5.66	4.19	3.09

To train the ANN, input and output data from a dataset were used to train the neural network according to the driver's behaviors during data acquisition. The vehicular test was conducted under different circuit conditions. In the city circuit, measurements were taken at various times of day and on different days of the week in Joinville. All experimental data were collected with the same driver to ensure consistent driving behaviors. For testing the artificial neural network, the weights generated during network training were applied.

CITY CIRCUIT

For data acquisition, the test vehicle followed urban routes, as illustrated in Fig. 3. The collected data were divided into two parts, creating two datasets. The first dataset was used to train the neural network, while the second dataset was reserved for testing, to verify the accuracy of the neural network after training.

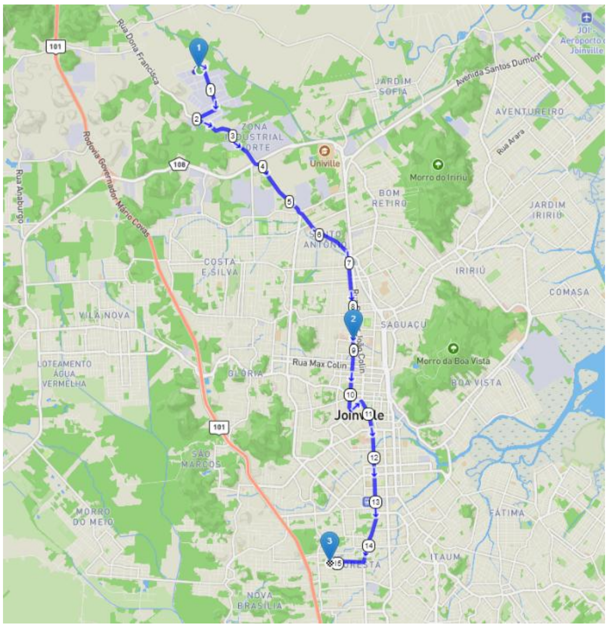

Fig. 3- Illustration of the city test route from north to south, passing through downtown. Segments 1 to 2 were used for training data, and segments 2 to 3 were used for validation data. (Map source: gpx.studio)

This route includes traffic lights and frequent traffic jams, especially during the downtown passage, necessitating frequent gear changes. However, the vehicle speed is constrained by city regulations. Fig. 4 illustrates the vehicle speed (left axis) and total gear ratio (right axis). In this scenario, the driver frequently changes gears, which is used to train the ANN.

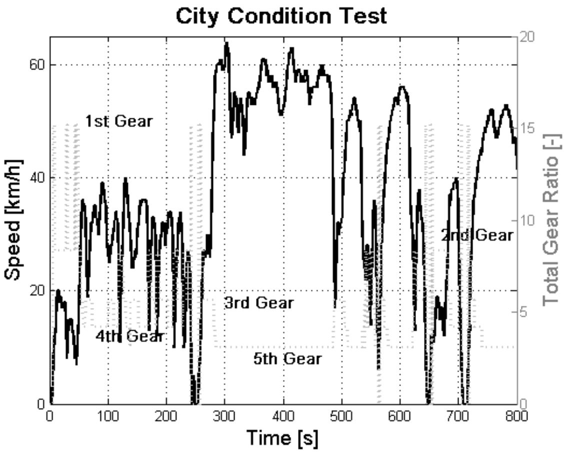

Fig. 4- Illustration of vehicle speed and selected total gear ratio during the city condition test

The experimental data were influenced by traffic flow, resulting in several time intervals with zero speed (when the gear was in neutral or the clutch pedal was pressed). To focus on relevant data for gear shifting analysis, instances with a speed value of zero were removed from the datasets. Several measurements (18 datasets) were taken on the city circuit to accumulate data from moments of gear change under different vehicle dynamic conditions. However, to include conditions of higher speed and greater engine torque demand, an intercity circuit following a highway route was also included in the training data.

INTERCITY CIRCUIT

Fig. 5 illustrates the route between the cities of Joinville and Curitiba, spanning approximately 138 km. This route features variable ground elevations, ranging from around 0 meters to 800 meters, and includes roads with significant slopes, some exceeding 6% (or 2.7°).

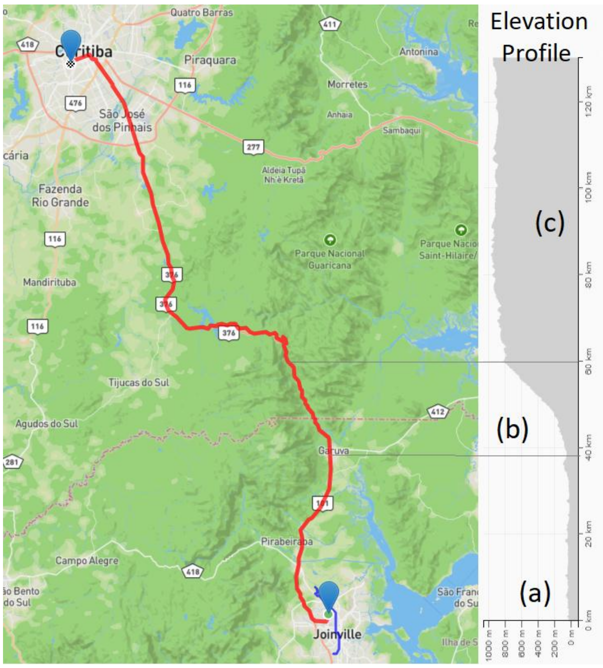

Fig. 5- Illustration of the intercity travel profile between Joinville and Curitiba. (a) Low elevation road, (b) Variable elevation road, (c) High elevation road. (Map source: gpx.studio)

The measured data from the intercity circuit, shown in Fig. 6, presents the engine load percentage as a function of vehicle speed and the percentage of accelerator throttle opening. In this scenario, the steep slopes in the mountain range increase the demand for more engine torque at high speeds. This situation is useful for mapping engine performance under severe conditions and serves as a basis for training the ANN under boundary conditions. The gear changes are illustrated in Fig. 7, where the vehicle is on the highway between 400 seconds and 5,300 seconds. The pattern of gear changes is presented in Fig. 8. The relationship between engine rotation and vehicle speed is constant for a specific gear, making it approximately a linear function. When the driver shifts gears, the driveline gear ratio is modified, and the slope of the linear function changes, as observed in Fig. 8. Clearly, at lower engine rotations, the relationship between vehicle speed and engine rotation becomes less distinct. In this case, the third variable – engine torque – is used to accurately identify the currently selected gear.

The data collected under both city and highway conditions were divided into two parts: approximately one half was used to train the ANN coefficients, while the other half was used to test the accuracy of gear selection based on driver behavior. The training set consisted of 24,655 gear patterns, while the test set included 19,652 gear patterns.

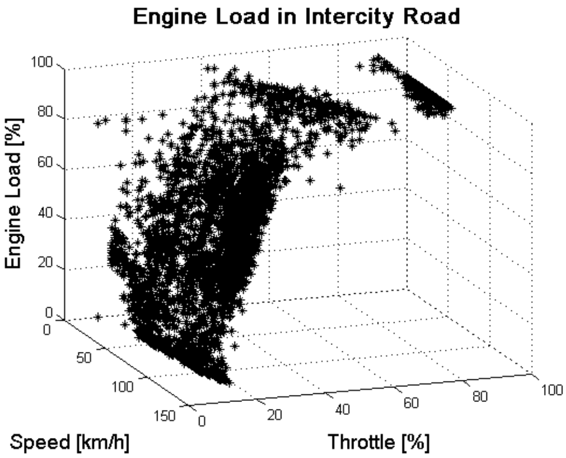

Fig. 6- Illustration of engine load in a plot of throttle percentage and vehicle speed during intercity travel

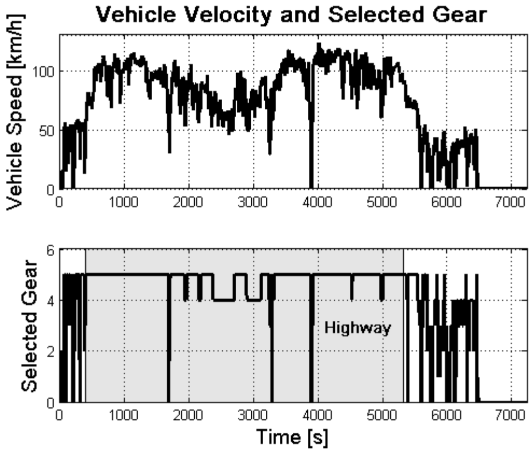

Fig. 7- Selected gear and vehicle speed during intercity travel

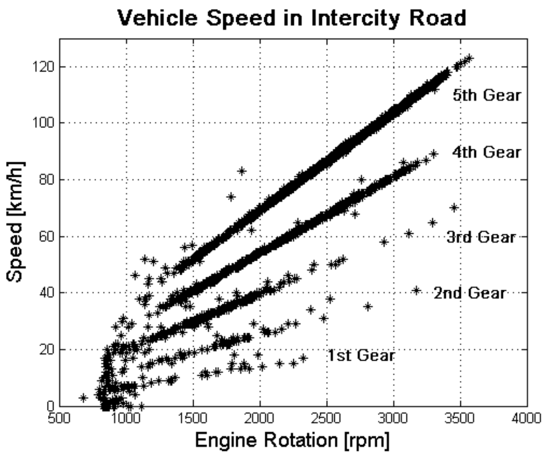

Fig. 8- Vehicle speed as a function of engine rotational speed during the intercity circuit

V. RESULTS

The ANN model training and validation process were divided into two parts. The first part involved a cross-validation evaluation using the K-Fold method. Once cross-validation was complete, the ANN was retrained on the entire dataset. Finally, the model was validated using a separate test set that was not involved in the cross-validation phase. This final validation provides an unbiased estimate of the model's performance on unseen data.

K-FOLD CROSS-VALIDATION

The K-fold cross-validation method involves dividing the dataset into K equal-sized subsets, where the model is trained and validated K times, each time using a different fold as the validation set and the remaining $K-1$ folds as the training set. This process is repeated for each fold. In this case, the training data were randomly split into 6 approximately equal-sized subsets, each containing an average of 6,117 samples ($K=6$), resulting in 6 iterations of training, as illustrated in Fig. 9.

Subsets	1	2	3	4	5	6
Run 1	Training Sets					Test
Run 2	Training Sets				Test	
Run 3	Training Sets			Test	Training Sets	
Run 4	Training Sets		Test	Training Sets		
Run 5		Test	Training Sets			
Run 6	Test	Training Sets				

Fig. 9- Diagram of K-fold cross-validation with $K=6$

After all 6 iterations, the performance metrics (accuracy and precision) from each validation round are presented in Table 2. The averaged values in the table show over 99% accuracy and a high precision (>0.99), meaning that, on average, the model correctly predicted the outcome 99% of the time, with consistent performance across all validation folds. This high accuracy indicates that the model is highly effective at correctly predicting gear selection within the driver-measured datasets. Therefore, the model is robust and demonstrates repeatability in training the ANN coefficients.

Table 2- K-fold cross-validation results for each iteration.

Run	Accuracy	Precision
1	99.58%	0.9924
2	99.32%	0.9910
3	99.41%	0.9787
4	99.79%	0.9926
5	99.92%	0.9971
6	99.78%	0.9934
Mean	99.63%	0.9909
Standard Deviation	0.0022	0.0063

As stable results were obtained during the cross-validation process, all the training data were used for a final training session, employing the same hyperparameters as those used during cross-validation. The determined coefficients were then applied to the test data for the final validation of the ANN.

VALIDATION WITH TEST DATA

The test data, illustrated in Fig. 10, is drawn from both city and intercity circuits and includes vehicle speed, engine rotation, and engine torque percentage. As previously mentioned, the relationship between these parameters for a particular gear becomes more evident when the engine rotation and vehicle speed are higher. Conversely, when both parameters are near their lower limits, the distinction between gears is less clear.

The Mean Squared Error (MSE) or Root Mean Square (RMS) error variation across each iteration is presented in Fig. 11. This metric measures the average squared difference between actual and predicted values throughout the training process. The process converges after approximately 140 iterations, with the RMS error falling below 0.004.

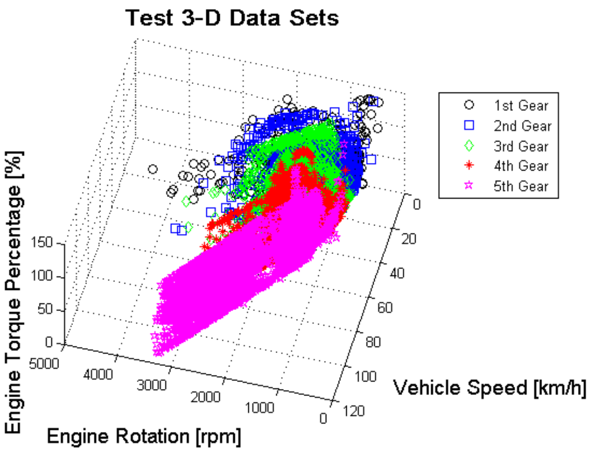

Fig. 10- 3D plot of the test datasets. Each gear category is illustrated with different markers and colors

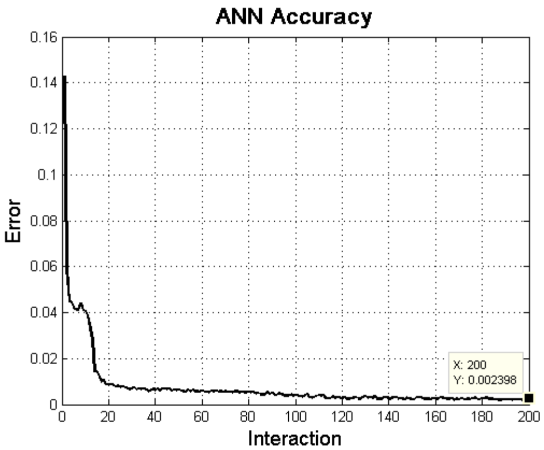

Fig. 11- RMS error during ANN training process

The trained ANN coefficients ($W_{ij}^1$and W_{jk}^2) correspond to the point of lowest error value in the weight space, as illustrated in Fig. 12. The error surface, plotted as a function of the weight coefficients, is generally non-linear and complex. By varying the coefficients around their trained values, the graph shows a convergence to a minimum error point, indicating effective training. This error function typically features a landscape with multiple local minima and saddle points, making the training process reliant on gradient descent or similar optimization algorithms to find this global minimum.

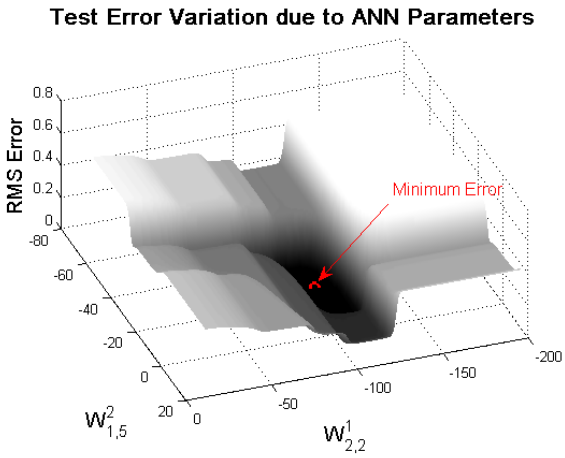

Fig. 12- Visualization of error function around converged values of $W_{2,2}^1$ and $W_{1,5}^2$

For the trained ANN coefficients, the ratio of correctly predicted gears to the total number of gear patterns in the test dataset was 99.64%. The obtained accuracy was determined by comparing the gear engaged by the neural network with the expected gear according to the experimental output data, as illustrated in Fig. 13. The patterns that did not match the actual selected gear (indicated as gear zero in the figure) were primarily related to clutch operation during a gear change. In certain traffic dynamics, the driver may keep the clutch disengaged for an extended period, disrupting the direct relationship between vehicle speed and engine rotation. These moments present a challenge in estimating a specific gear, as the driveline is uncoupled from the engine.

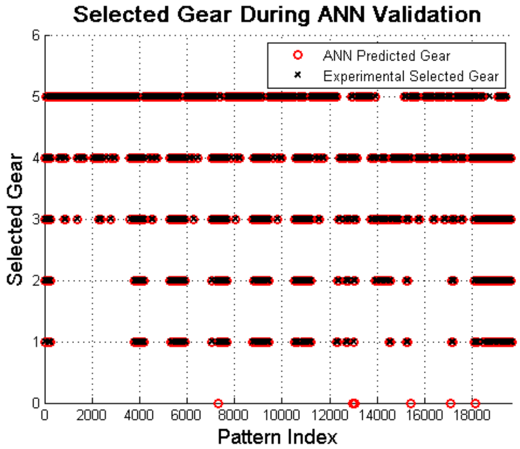

Fig. 13- Predicted gear by the ANN compared to the actual gear selected during the vehicular test

Table 3 presents a confusion matrix that illustrates the performance of the trained ANN. The matrix displays true positives, true negatives, false positives, and false negatives for each gear. In this context, a true positive, located along the matrix's principal diagonal, represents an estimated gear that matches the actual selected gear. A false negative occurs when the ANN fails to correctly predict the actual gear, indicated by elements below the principal diagonal, while a false positive is indicated by elements above the principal diagonal. The upper numbers within the matrix represent the count for each category, and the numbers below them indicate the percentages relative to the total occurrences of the actual gear. The results in Table 3 show that the model's accuracy is relatively high for each selected gear. For all five estimated gears, the percentage of correct identifications (true positives) exceeded 99.5%.

Table 3- Confusion matrix for the estimated gear using test datasets.

		Actual Values				
		1st	2nd	3rd	4th	5th
Predicted Values	1st	736 99.7%	1 0.1%	0 0.0%	0 0.0%	0 0.0%
	2nd	2 0.3%	1385 99.9%	0 0.0%	0 0.0%	0 0.0%
	3rd	0 0.0%	0 0.0%	2681 99.9%	20 0.5%	0 0.0%
	4th	0 0.0%	0 0.0%	0 0.0%	3731 99.5%	13 0.1%
	5th	0 0.0%	0 0.0%	0 0.0%	0 0.0%	11077 99.9%

From the confusion matrix, the precision is calculated as the ratio of correctly predicted positive observations to the total predicted positives, as shown in Table 4. All gear estimations have a precision exceeding 99.2%. For recall (sensitivity), which measures the ratio of correctly predicted positive observations to all observations in the actual class, the estimated gear also exceeds 99%. The F1-Score, a harmonic mean of precision and recall, also provided values over 99% (Table 4). These results indicate that the model is highly effective for controlling gear changes in automated gearbox vehicles.

Table 4- ANN performance metrics based on true positives, true negatives, false positives, and false negatives for each estimated gear.

Gear	Precision	Recall	F1-Score
1st	0.9986	0.9973	0.9980
2nd	0.9986	0.9993	0.9989
3rd	0.9926	1.0000	0.9963
4th	0.9965	0.9947	0.9956
5th	1.0000	0.9988	0.9994
Mean	0.9973	0.9980	0.9976

For an overall metric, the macro-average was calculated by taking the unweighted mean of the precision, recall, and F1-score across all classes, as shown in Table 4. These mean values indicate that the model effectively minimizes false positives (precision) and false negatives (recall), demonstrating a balanced performance between precision and recall. This balance highlights the model's robustness in selecting the appropriate gear under varying vehicle dynamic conditions.

VI. CONCLUSIONS

This research implemented a Perceptron-based Artificial Neural Network (ANN) to optimize gear shifting in automated transmissions, with a focus on the dynamics of a conventional powertrain. The results demonstrate that the proposed ANN is highly effective in learning and replicating driving patterns, achieving a gear selection accuracy of 99.64% across various dynamic vehicle conditions. The neural network's robustness and precision were further supported by the confusion matrix, where true positive rates for gear prediction exceeded 99.5% across all gears, indicating the model's reliability in real-world scenarios. An advantage of this approach is its low computational cost. The simplicity of the Perceptron model ensures that it requires significantly fewer computational resources compared to more complex neural networks, while still delivering high performance. The training process converged with a root mean square (RMS) error below 0.004 after approximately 140 iterations, highlighting both the speed and efficiency of the model. This makes it particularly well-suited for real-time applications, where processing speed and resource efficiency are critical. The balance between high accuracy and low computational demand underscores the model's potential for widespread integration into automotive control systems, offering a scalable and effective solution adaptable to various vehicle types. The study's findings open new possibilities for enhancing vehicle performance through intelligent control strategies. By reducing the physical demands on drivers and improving the overall driving experience, this approach aligns with the broader goals of modern automotive innovation. Future research can build on these results by integrating this Perceptron-based method with other vehicle control systems, further advancing the development of smarter, more efficient, and user-friendly vehicles.

REFERENCES

[1]. Hailu, H.N., Redda, D.T.: Design and development of power transmission system for green and light weight vehicles: a review. The Open Mechanical Engineering Journal. 12, 81-94 (2018).

[2]. Kim, W.-S., Kim, Y.-J., Kim, Y.-S., Baek, S.-Y., Baek, S-M., Lee, D.-H., Nam, K.-C., Kim, T.-B., Lee, H.-J.: Development of control system for automated manual transmission of 45-kw agricultural tractor. Appl. Sci., 10, 1–21 (2020).

[3]. Abediasl, H., Ansari, A., Hosseini, V., Koch, C.R., Shahbakhti, M.: Real-time vehicular fuel consumption estimation using machine learning and on-board diagnostics data. Proceedings of the Institution of Mechanical Engineers, Part D: J. Automobile Engineering. 0(0), 1–14 (2023).

[4]. Hwang, M.H., Lee, G.S., Kim, E., Kim, H.W., Yoon, S., Talluri, T., Cha, H.R.: Regenerative braking control strategy based on AI algorithm to improve driving comfort of autonomous vehicles. Appl. Sci. 13(946), 1–96 (2023).

[5]. Lv, C., Xing, Y., Lu, C., Liu, Y., Guo, H., Gao, H., Cao, D. Hybrid‐learning‐based classification and quantitative inference of driver braking intensity of an electrified vehicle. IEEE Transactions on Vehicular Technology, 67(7), 5718–5729 (2018).

[6]. Kwon K., Lee, J.-H., Lim, S-K.: Optimization of multi-speed transmission for electric vehicles based on electrical and mechanical efficiency. Applied Energy. 342, 1–14 (2023).

[7]. Shukla, A., Sharma, R.: Neural network based variable DC-link voltage control of electric vehicle driveline. Smart Science. 12(2), 319–342 (2024).

[8]. Kwon, K, Lim, S.-K., Kim, D., Park, K.: Automation program for optimum design of electric vehicle powertrain systems based on artificial neural network. eTransportation. 18, 1–13 (2023).

[9]. Yang, R., Yan, Y., Sun, X., Wang, Q., Zhang, Y., Fu, J., Liu, Z.: An artificial neural network model to predict efficiency and emissions of a gasoline engine. Processes, 10(204), 1–20 (2022).

[10]. Kanchev, H., Hinov, N., Gilev, B., Francois, B.: Modelling and control by neural network of electric vehicle traction system. Elektronika IR Elektrotechnika. 24(3), 23–28 (2018).

[11]. Park, J., Cho, J., Choi, H., Park, J.: Prediction of reformed gas composition for diesel engines with a reformed EGR system using an artificial neural network. Energies. 13, 1–17 (2020).

[12]. d'Apolito, L., Hong, H.: Forklift truck performance simulation and fuel consumption estimation. Journal of Engineering, Design and Technology. 18(3), 689–703 (2020).
[13]. Seo, J., Yun, B., Parka, J., Park, J., Shin, M., Park, S.: Prediction of instantaneous real-world emissions from diesel light-duty vehicles based on an integrated artificial neural network and vehicle dynamics model. Science of the Total Environment. 786, 1–12, (2021).
[14]. Saeed, A.B.: Simulation design of an intelligent system for automotive transmission gearbox based on FPGA. International Journal of Engineering Technology. 6(2), 395–411 (2018).
[15]. Petrescu, R.V.V., Aversa, R., Apicella, A., Petrescu, F.I.T.: Dynamic synthesis of a classic, manual gearbox. American Journal of Engineering and Applied Sciences. 11(2), 586–597 (2018).
[16]. IBGE, Brazilian Institute of Geography and Statistics: Frota de veículos (Vehicle fleet). IBGE. https://cidades.ibge.gov.br/brasil/pesquisa/22/28120 (2024). Accessed 19 August 2024.
[17]. Aggarwal, C.C.: Neural Networks and Deep Learning: a Textbook. Springer, Berlin (2018).
[18]. Haykin, Simon. Neural Networks: A Comprehensive Foundation. Prentice Hall, New Jersey (1999).