

# Anomaly Detection in Traffic Trajectories Using a Combination of Fuzzy, Deep Convolutional and Autoencoder Networks\*

Research Article

Mojtaba Banifakhr<sup>1</sup>

Mohammad Taghi Sadeghi<sup>2</sup>

**Abstract:** Due to the increasing deployment of vehicles in human societies and the necessity for smart traffic control, anomaly detection is among the various tasks widely employed in traffic monitoring. As the issue of urban traffic and their relative smart monitoring systems have gained popularity among researchers in recent years, there exist several studies in this regard. In most of these studies, classification is performed based on the behavior of drivers, where a set of default trajectories are used in order to learn the system and classify the related data. However, two understudied challenges are the lack of access to sufficient data to provide an efficient model, along with the lack of access to anomaly data that covers all possible abnormal trajectories. While the former challenge can be tackled through long-term data recording, the latter requires appropriate considerations. To this aim, we have utilized a combination of optimized convolutional neural network and fuzzy neural network classifiers, along with autoencoding neural networks. The final combination occurs at the decision level. First, the CNN-ANFIS classifier assigns the input trajectory to one of the predefined categories. Then, the trained autoencoder networks examine the result in order to find whether the trajectory is normal or abnormal. Obtaining 87.5% accuracy on QMUL and 99.5% on the T15 datasets confirms the superior performance of the proposed method.

**Keywords:** Adaptive Neural-Fuzzy Inference System, Autoencoder Neural Network, Deep Neural Network, Trajectory Anomaly Detection

## 1. Introduction

In recent years, advancements in software and hardware technologies and development of innovative machine learning algorithms and advanced processing tools have enabled us to employ these technologies in various fields of smart detection [1]. An application of artificial intelligence that can effectively prevent accidents in urban areas (associated with considerable casualties every year) is the analysis and detection of anomalies on the vehicle traffic trajectories.

The dramatic increase in the number of vehicles in urban and intercity passages and the consequent increase in the number of accidents caused by traffic violations have increased the need for an efficient surveillance system. Therefore, in recent years, the utilization of video surveillance systems for online traffic management and traffic safety of the roads has drawn much attention of researchers in many fields. By employing a smart system, in addition to the capability to control appropriate driving behaviors and monitor the trajectories of vehicles, the

detection of high-risk driving behaviors can be carried out with acceptable accuracy. Analyzing and monitoring surveillance videos by a human operator increase costs, data storage memory, and the possibility of human error. Hence, to overcome these problems, intelligent methods for detection of accidents and anomalies should be used based on high-speed and low-cost video surveillance to assist traffic operations and provision of law. As a result, the existence of effective methods to identify vehicle behavior that provides automatic detection of anomalies in trajectories using surveillance videos is crucial. However, automated training of a system to comprehend vehicle behaviors using surveillance videos is extremely challenging, carried out generally in three stages: extracting vehicle information, presenting this information, and understanding vehicle behavior.

In this study, a novel method is presented to detect anomalies in trajectories from video surveillance sequences. The method employs artificial intelligence and deep learning techniques to deal with the aforementioned challenges.

Employing this smart video surveillance system is expected to provide online and high-accuracy control of vehicles by monitoring their trajectories and detecting high-risk behaviors. To this aim, a proper representation of input data to obtain the best possible model is among the issues that should be considered. In this framework, Convolutional Neural Network (CNN) seems a very good choice [2]. Moreover, high flexibility in case of high diversity in the input data is another significant component, which can be fulfilled using Adaptive Neuro-Fuzzy Inference System (ANFIS) [3]. Finally, a structure based on Autoencoder Neural Networks (AE) [4] can be a good choice to exploit datasets in different classes to provide unsupervised anomaly detection. Consequently, this paper proposes an efficient structure using a combination of Deep Neural Networks (DNNs) [2], Adaptive Neuro-Fuzzy Inference System (ANFIS) [3], and Autoencoder Neural Networks (AE) [4] to provide effective and automatic monitoring of traffic.

There exist many challenges in detecting traffic trajectory anomalies using video sequences. The first challenge is the need for a large labeled training dataset for the training process. Moreover, due to the high diversity of data, the other challenge is the need for a high level of perception and understanding the system (i.e., proper flexibility) beyond the learning process with training datasets to detect sample data properly and with good accuracy. Because of these challenges, classic methods used in this field do not generally yield satisfactory results. In this study, to overcome these problems, a method is proposed that, in addition to training a deep model with extended training

\* Manuscript received, July, 9, 2021; accepted, October, 7, 2021.

<sup>1</sup> Ph.D. Student in Communication Engineering, Azadi Campus, Yazd University, Yazd, Iran.

<sup>2</sup> Corresponding Author: Associate Professor, Department of Electrical Engineering, Yazd University, Yazd, Iran.  
Email: m.sadeghi@yazd.ac.ir.

datasets, provides the capability to add rules similar to human perception and understanding (known as fuzzy rules). Moreover, in order to more generalize the classifier and ignore insignificant features, the proposed structure makes use of Autoencoders (AE).

Hence, the proposed system uses a neuro-fuzzy classifier in order to provide the necessary flexibility to deal with unencountered events. In addition to flexibility, the proposed system should possess another feature to ensure proper performance: exploiting the training data in the best possible manner to extract efficient features. For this aim, DNNs and AEs are among the best options [4]. In addition to the possibility of obtaining higher-level features using training data, DNNs provide classification capability. Moreover, AE networks are effective tools to detect the compatibility of new training data with the current training data.

The other specific approach considered in this study is the optimization of hyperparameters in DNNs using the Whale Optimization Algorithm (WOA). This optimization aims to improve the classification results by offering an architecture with optimized hyperparameters [5].

Fuzzy algorithms are utilized in a wide range of studies. Most of these studies utilize the proper performance and high flexibility of Fuzzy algorithms in processes where data is highly diverse. For instance, the ANFIS algorithm is employed in [3] to classify customer credits into good and bad classes. The implementation of this algorithm on the Standard German Credit dataset indicates its proper performance, which is confirmed by the high accuracy percentage in the classification of customers.

Regarding the studies on anomaly detection using trajectory data, one can refer to [6], where trajectories are determined through analysis of cellphones of the drivers, and a process for anomaly detection is implemented. In this study, anomaly detection is carried out by analyzing the trajectory data. Various features of the users in specific timeframes are selected as criteria to determine normal or abnormal behaviors. The features employed in this study include the traveled distance, average speed, and arithmetic average speed. These features are collected in a dataset with 6853 real-world trajectories. Obtaining 98 percent detection accuracy indicates its satisfactory performance in the anomaly detection process.

Using hierarchical clustering, researchers detected anomalies in real-world video sequences in [7] based on similarities among trajectories. In this paper, results are evaluated using the Fuzzy K-means algorithm. The proposed method in this article demonstrates good performance in detecting abnormal trajectories, even in the presence of noise or high traffic. Moreover, in [8], a novel method to detect anomalies based on fuzzy theory is presented that demonstrates proper performance in a diverse set of conditions, including different light, weather, and traffic congestion. In this method, the fuzzy theory is employed in preprocessing, trajectory extraction, and anomaly detection to offer a practical approach in anomaly detection.

One major challenge in anomaly detection in video sequences from surveillance cameras is complex incidents in scenes with a high number of vehicles. A suitable solution to overcome this challenge is a novel learning method based on the fuzzy transfer learning neural network platform, as proposed in [9].

In addition to the methods above, the utilization of DNNs

in detecting anomalies has shown promising results. For instance, [10] evaluates employing deep Convolutional Neural Networks (DCNNs) on the extracted temporal-spatial components. As indicated in the results of this study, the proposed method offers good performance in crowded video scenes.

The use of autoencoder networks to detect anomalies has a long history. For instance, in [11], an autoencoder network based on LSTM was employed to detect anomalies in two datasets with point anomalies. The method proposed in this study employed nonlinear layers to extract innate features stashed in data and thus, demonstrated superior performance compared to other classical pattern recognition methods.

In [12], a Decision-Tree enabled approach using deep learning was proposed to extract anomalies from traffic cameras while accurately estimating the start and end times of the anomalous event. Their approach included creating a detection model based on YOLOv5. The anomaly detection and analysis step entailed traffic scene background estimation, road mask extraction, and adaptive thresholding. Candidate anomalies were passed through a decision tree to detect and analyze final anomalies.

In [13], a deep learning-based feature visualization method was proposed to map 3-dimensional features into a RGB color space. A color trajectory was then derived by encoding a trajectory with the RGB colors. The spatial and temporal properties was extracted from the trajectories. Then, GIS map fusion is conducted to obtain insights for better understanding the traffic anomaly locations, along with the influences on the road affected by the corresponding anomalies.

In [14], various automatic and real-time surveillance methods were addressed for abnormal event detection to recognize the dynamic crowd behavior in security applications. This study classified methods into different categories such as tracking, classification based on handcrafted extracted features, classification based on deep learning, and hybrid approaches. Hybrid and deep learning methods demonstrated better results in the classification stage.

In [15], each video is represented as a group of cubic patches for identifying local and global anomalies. A unique sparse de-noising autoencoder architecture is to reduce the computation time and improve results. Experimental analysis on two benchmark data sets – the UMN dataset and UCSD Pedestrian dataset - confirmed that the algorithm proposed in this study outperforms the state-of-the-art models in terms of false positive rate, while showing a significant reduction in computation time.

Finally, [16] presented an efficient and robust method for solving unsupervised traffic anomaly detection based on vehicle trajectories. In this study, possible anomalies were detected and tracked from the background image sequence of videos. The start time of the abnormal events is located by the decision module based on tracks.

The remainder of this article is organized as follows. In Section 2, the fundamental elements of the proposed method are reviewed. Section 3 evaluates the proposed method in three parts: database, system training, and system evaluation. Simulation results are presented in Section 4, and Section 5 summarizes the conclusions of the study.

## 2. The Fundamentals of Research

To establish a proper context for the proposed method, the following section presents the employed structures in this method. The first investigated structure in this section is deep Convolutional Neural Networks (DCNNs).

### 2.1. Deep Convolutional Neural Networks

Deep Convolutional Neural Networks (DCNNs) are among the most widely employed structures in recent studies on pattern recognition and machine learning. In these networks, various layers are defined that each possesses a specific function according to coherent principles. In general, a CNN consists of three basic layers that possess a specific function: the Convolution Layer, Pooling Layer, and Fully Connected Layer. Figure 1 shows a general structure of a CNN.

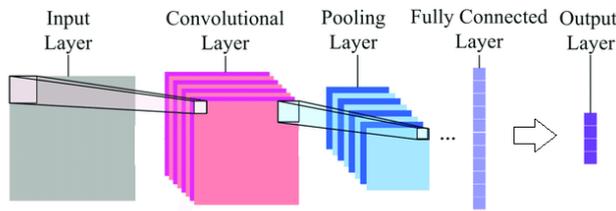


Figure 1. General Structure for DCNN [12]

In CNNs, we deal with two classes of parameters and hyperparameters. While system parameters are trained in feedforward and backpropagation processes, hyperparameters (which are crucial) cannot be determined using conventional methods. In this study, to improve the performance of our DCNN, the optimization of the three significant hyperparameters of the learning rate, momentum rate in gradient descent process, and regularization norm are considered.

#### 2.1.1. Learning Rate and Momentum

The utilization of the Momentum Algorithm in the Stochastic Gradient Descent (SGD) framework is among the best and most well-known approaches to update weights in neural networks. In this approach, the weight matrix ( $W$ ) of the network is updated via a linear combination of the negative gradient of the cost function  $-\nabla L(W)$  and the weight changes in the previous step ( $V_t$ ) as follows:

$$V_{t+1} = \mu V_t - \alpha \nabla L(W_t + \mu V_t)$$

$$w_{t+1} = w_t + V_{t+1} \tag{1}$$

where  $\alpha$  is the learning rate that determines the effect of the gradient value on the updating process, and  $\mu$  is the momentum value that determines the weight for the previous update.

In general, Equation (1) is employed to determine the new value for the change in parameter vector  $V_{t+1}$  (and consequently  $W_{t+1}$ ) in iteration  $t+1$  using the value  $V_t$  for the previous update and the current weight matrix  $W_t$ .

#### 2.2.2. Regularization norm

In deep learning processes, the general objective is

minimizing the following cost function:

$$J(w^{[1]}, b^{[1]}, \dots, w^{[L]}, b^{[L]}) = \frac{1}{m} \sum_{i=1}^m L(\hat{y}^{(i)}, y^{(i)}) \tag{2}$$

In this equation,  $L$  is an arbitrary cost function indicating the classification error. To improve the classification results, regularization techniques are used as an efficient tool. To this aim, another term is added to adjust the weights, as shown in the following equation:

$$J(w^{[1]}, b^{[1]}, \dots, w^{[L]}, b^{[L]}) = \frac{1}{m} \sum_{i=1}^m L(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m} \sum_{i=1}^L \|w^{(L)}\|_F^2 \tag{3}$$

where  $\lambda$  is the weight of the regularization term (i.e., the parameter that controls the importance of the regularization term), and  $F$  refers to the Frobenius norm, equal to the square of the matrix norm. Proper determination of  $\lambda$  is a crucial issue. This parameter can be adaptively selected according to the condition.

### 2.2 Whale Optimization Algorithm

Whale Optimization Algorithm (WOA) is a new and efficient optimization algorithm [5]. The algorithm is not expensive in terms of computational complexity. Moreover, the algorithm has a good convergence ability in both complex and simple objective functions. These advantages encouraged us to employ this algorithm for optimizing the aforementioned hyperparameters of CNN. The adopted optimization process is presented in detail in Section 3.B.

### 2.3. Autoencoder Neural Network

Autoencoder (AE) neural networks are among the most widely adopted neural network structures for a broad range of pattern recognition applications, including classification, clustering, feature compression, and data reconstruction. Figure 2 shows the general structure of these networks.

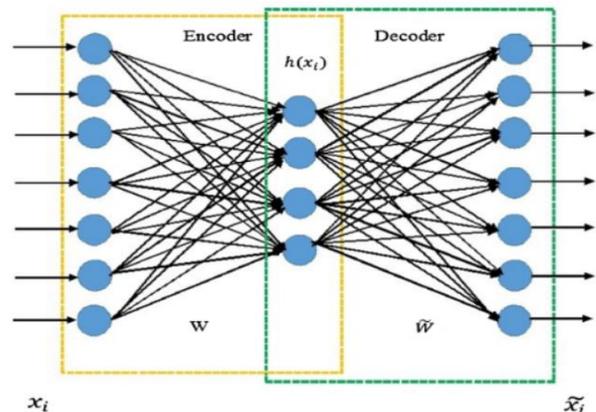


Figure 2. The structure for Autoencoder Networks [17]

As Figure 2 shows, input ( $x_i$ ) and output ( $\hat{x}_i$ ) have a similar dimension while  $h(x_i)$  has a different dimension. Generally,  $h(x_i)$  is in a lower dimension than others. You can find more information in [4].

Compared to other neural networks, AE networks offer excellent results in unsupervised conditions, which can be

utilized in this work. An AE is trained in order to adjust the output to the input as much as possible. Thus, an advantage of autoencoder networks is that they are forced to generalize data and seek common patterns from training data. Therefore, after training such a network, by comparing the reconstructed data in the output to the input data using specific criteria, we can determine whether the given data can be classified as a member of the associated class or not.

#### 2.4. ANFIS-Based Classifier

Adaptive Neuro-Fuzzy Inference System (ANFIS) networks can be considered among the most efficient fuzzy inference systems. Since such systems employ fuzzy rules, they are similar to fuzzy systems, and they are neural networks because they are trained as one. In other words, ANFIS is a trainable network with functionality very similar to a fuzzy inference system and advantages of neural networks. Here,  $x$  and  $y$  are presumed as inputs of the desired network, and  $z$  as its output variable [18]. Now, if the rules are as follows:

Rule1: if  $x$  is  $A_1$  and  $y$  is  $B_1$  then  $f_1 = p_1x + q_1y + r_1$

Rule2: if  $x$  is  $A_2$  and  $y$  is  $B_2$  then  $f_2 = p_2x + q_2y + r_2$

(4)

and if the center average defuzzifier is employed as the defuzzifier, the equivalent structure of ANFIS will be as shown in Figure 3:

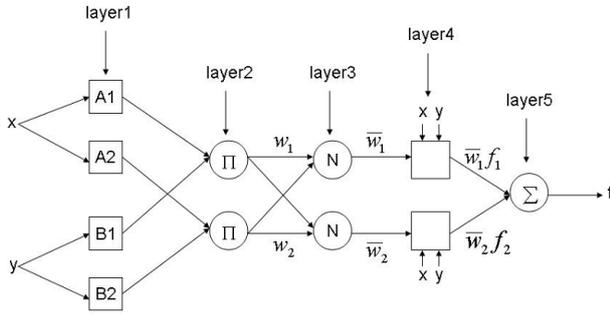


Fig 3. ANFIS Network Structure.

In the following, we will describe the selected layers.

Layer 1: In this layer, the inputs pass through the Membership Functions (MF).

$$O_{1,i} = \mu A_i(x), \text{ for } i = 1,2 \quad (5)$$

$$O_{1,i} = \mu B_i(x), \text{ for } i = 3,4 \quad (6)$$

For each function, membership functions are generally selected as Gaussian. For instance, the Standard Normal Distribution:

$$\mu A(x) = \frac{1}{1 + \left| \frac{x-c_i}{a_i} \right|^{2b_i}} \quad (7)$$

where  $\{a_i, b_i, c_i\}$  are the set of parameters. The components of this layer are known as the primary components.

Layer 2: The output in this layer is the multiplication of input signals, equivalent to the ‘‘iF’’ section of the rules.

$$O_{2,i} = w_i = \mu A_i(x)\mu B_i(y), i = 1,2 \quad (8)$$

Layer 3: The output in this layer is the normalized value of its previous layer:

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2}, i = 1,2 \quad (9)$$

Layer 4: The output of this layer is as follows:

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i(p_i x + q_i y + r_i) \quad (10)$$

Layer 5: Finally, the output of this layer is the output of the system:

$$O_{5,i} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (11)$$

The accumulation of the aforementioned structure forms the ANFIS network employed in this study. According to the results of similar researches on classifications based on this structure, a high accuracy percentage is expected.

In the following section, the proposed method, and the desired system structure for detecting anomalies are presented.

### 3. The Proposed Method

In section 2, the crucial and beneficial components of the proposed method were thoroughly investigated. In this section, within the framework of anomaly detection, the proposed process that utilizes a combination of the above mentioned three parts is presented. Figure 4 demonstrates the block diagram for the proposed method.

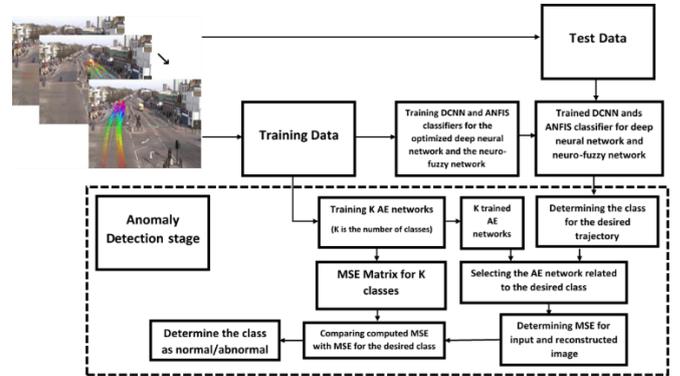


Figure 4. The block diagram for the proposed method

The block diagram for the proposed method includes multiple parts that will be described in the following sections.

#### 3.1. Training the Combined Classifier Using CNN and ANFIS

In the training process utilized in our proposed method, the first section is training the DCNN and ANFIS classifier. Ultimately, it is aimed that the desired anomaly can be detected using these trained networks and the autoencoder network.

The training process of the classifier for the optimized CNN and ANFIS network shown in Figure 4 can be seen in the block diagram of Figure 5.

In this process, once the appropriate training data is

provided, the data augmentation process is employed to enhance the training process of the deep CNN. To this aim, a Generative Adversarial Network (GAN) [19] is used. The number of generated trajectories to enhance the learning of QMUL and T15 datasets are 140 and 900, respectively, resulting in a total of 256 training data for the first dataset and 1950 for the second.

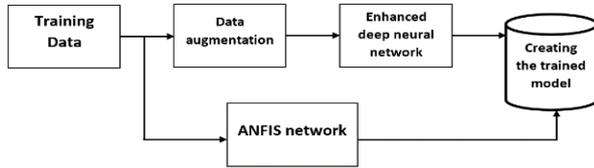


Figure 5. CNN and ANFIS training process

As mentioned before, the enhanced DCNN classifier is employed to acquire a proper model from the input data. Moreover, since the abnormal activities are usually highly diverse, the neuro fuzzy-based classifier is used to provide more flexibility in the classification process. In the following section, we first examine the utilized enhanced DCNN and its optimization process. Then, we evaluate the classifier based on the ANFIS.

### 3.2. The Enhanced Deep Convolutional Neural Network

As mentioned in previous sections, the appropriate values for the initial learning rate, momentum, and L2 regularization norm hyperparameters have to be optimized using an appropriate optimization process. In this section, the adopted optimization process is presented. Figure 6 demonstrates the block diagram of the optimization algorithm.

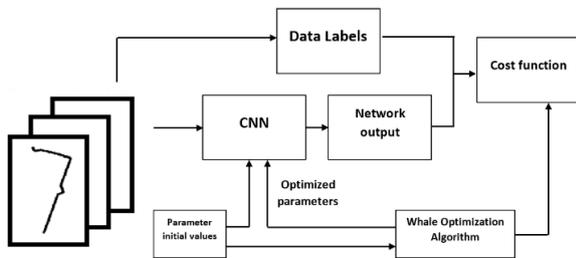


Figure 6. The structure for the optimization of the CNN

In the optimization process, random values (in specific ranges) are assigned for the initial value of the hyperparameters to form the initial CNN. Then, using these values, the network is trained and the trained network is implemented on the validation dataset. The cost function is then calculated, and the hyperparameters are updated using the WOA in each iteration. In this process, the population size of 50, maximum number of iteration of 300, and nonlinear convergence factor of 1 were selected. Cost function is reduced in each iteration to obtain the best values. Once the cost function is not changed in two separate iterations, the optimization process is finished. The results for learning rate, momentum, and L2 regularization norm hyperparameters provide an efficient CNN.

### 3.3. Training the Deep Autoencoder Network

In the next step of the proposed method, a total of K

autoencoder networks (K indicates the number of classes in the considered dataset) are independently trained using the exclusive training dataset for each class. The output of this stage is a total of K trained deep autoencoder networks in the first step and a reconstruction error matrix for each class in the next. These two outputs help us improve the capability of our method in the anomaly detection process.

### 3.4. Anomaly Detection by Combining the Subsections

At the final stage of our proposed method, the data is diagnosed for anomalies using the trained CNN+ANFIS and autoencoder network. The procedure for this task is as follows: first, the nearest class for the new data is determined using the trained combined CNN and ANFIS classifier. Then, the trained autoencoder associated with this class is used, the data is fed to this autoencoder, and the reconstruction mean square error (MSE) value is determined. If this value is within the defined range for the MSE of the desired class of the deep autoencoder network, the data is categorized as normal. Otherwise, it is anomalous. This process is shown in the lower part of Figure 4.

### 3.5 Validation

In studies related to machine vision and pattern recognition, a wide range of criteria are employed for model validation and evaluation of the results. One of the most widely employed criteria to evaluate the classification process is accuracy. Using this criterion, the performance of the classifier can be assessed simultaneously on both normal and abnormal data.

Accuracy is the most common, fundamental, and simplest criterion to evaluate the quality of a classifier, which is the extent of correct detection of the classifier in the accumulation of the two categories. This criterion demonstrates the number of patterns detected correctly. Based on the matrix provided earlier, it is formulated and defined as (12).

$$\text{Accuracy} = \frac{TN+TP}{TN+FP+FN+TP} \quad (12)$$

where TP, FP, TN, and FN are True Positive, False Positive, True Negative, and False Negative, respectively.

To provide a better evaluation of our proposed approach, a 10-fold cross-validation technique is used. For this purpose, total data for training and validation is divided into ten parts. Nine parts (equivalent to 90 percent of the data) are used for training, and the remaining 10 percent are employed for the test. This process continues until all the data are employed in the testing process. This approach is known as the 10-Fold Cross-Validation. Following the examination of the proposed method and evaluation criteria, we shall seek simulation and the output data to enable the final analysis of the proposed system.

## 4. Simulation

In this section, our experiments and simulations results are presented.

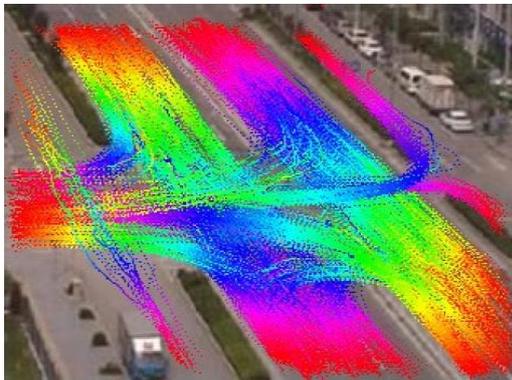
### 4.1. Input Data

One of the crucial parts of any study is the dataset employed for method evaluation. In this research, the datasets used for

the evaluation of the proposed method are the well-known QMUL and T15 datasets. Figure 7 shows some trajectories from these datasets and Table 1 shows the details of these two datasets.



(a)



(b)

Figure 7. The datasets employed in this study: a) QMUL[20], b) T15 [21]

Table 1. The details of the datasets [4]

Dataset	Total Number of Trajectories	Number of Classes	Number of abnormal trajectories
QMUL[20]	166	7	17
T15[21]	1531	15	31

As we can see in Table 1, the QMUL dataset includes 166 trajectories (149 normal trajectories and 17 abnormal ones). The data in this dataset can be categorized into seven classes. In Figure 8, some of the trajectories in this dataset for each class are demonstrated.



Figure 8. QMUL dataset [4]

The second dataset utilized in this study is T15. This dataset includes 1531 trajectories categorized into normal trajectories with 1500 samples and 31 abnormal trajectories. Moreover, normal trajectories in this dataset are organized into 15 different subcategories with 100 trajectories in each. Figure 9 shows some of the images in this dataset.

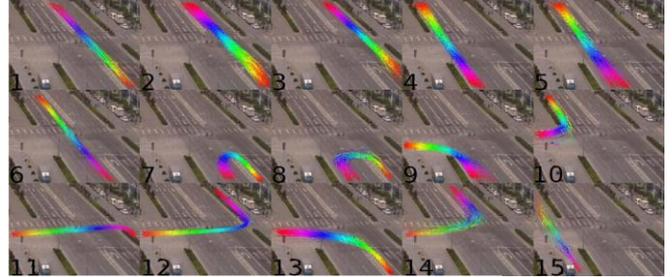


Figure 9. T15 dataset [4]

It is essential to mention that the images in these two datasets are captured using traffic surveillance cameras, where the camera is fixed. In other words, the captured images are registered.

#### 4.2. Convolutional Neural Network

For the simulation of the proposed method, a deep neural network is employed. In the following section, the layers of this deep neural network are described.

Prior to presenting the specifications of the CNN network, we should note that the inputs to this network are all resized to 168\*168 pixels. The employed network is a modified ResNet12 network, which demonstrated exceptional performance in [20, 21]. We utilized ResNet12 as the basis for our proposed CNN network. This network consists of 4 residual blocks, each with 3 convolution layers and 3\*3 kernels. Moreover, one 2\*2 max-pooling layer is applied after every 3 blocks and a global pooling layer after the fourth. Furthermore, dropout is employed in our proposed structure, and the number of filters is changed from (640, 320, 160, 64) to (512, 256, 128, 64). These changes are applied to improve the results and achieve better performance.

As for the configuration for the training of the network,  $\alpha$  is the initial learning rate, the training rule is based on gradient descent with a momentum of  $\mu$ , the number of iterations is 50, and the regularization norm (R) is 0.1.

The number of training data in the first dataset is 256 trajectories in 7 classes, along with 1950 trajectories in 15 classes in the second dataset. The values for initial learning rate, momentum, and L2 regularization norm are optimized in every iteration to yield the highest optimization level. In simulations, the optimum values for the first dataset were 0.0121 for  $\alpha$ , 0.9853 for  $\mu$ , and 0.0035 for R. In addition, in the second dataset, the optimized value for  $\alpha$  was 0.0105, while  $\mu$  was 0.9794, and R yielded 0.0029.

#### 4.3. ANFIS Network

The ANFIS network employed in this study is a Sugeno network [18]. In our adopted structure, a triangle function with RMSE=0.1635,  $R^2=0.96$ , and 20 rules was considered. The number of membership functions utilized was 7 and 15, and the number of iterations was 50.

**4.4. Autoencoder Network (AE)**

The autoencoder network used in this study is similar to the one in [17]. This network consists of two parts: encoder and decoder. The input for the encoder part  $q_{\theta}(z|\tau)$  is  $\tau$ ,  $z$  is its output, and  $\theta$  indicates the weights and biases for the encoder network. Moreover, the decoder  $p_{\phi}(\tau|z)$  receives  $z$  as input and produces  $\tau$  as output, while  $\phi$  indicates the weights and biases for the decoder.

Equation 13 illustrates the calculation of error function  $l_i$  for the trajectory  $\tau_i$ .

$$l_i(\theta, \phi) = -ll + KLD \tag{13}$$

In this equation, simulation ( $ll$ ) is determined using (14), while Kullback-Leibler Divergence (KLD) is calculated according to equation (15) [22].

$$ll = E_z \sim q_{\theta}(z|\tau_i)[\log p_{\phi}(\tau_i|z)] \tag{14}$$

$$KLD = q_{\theta}(z|\tau_i)||p(z) \tag{15}$$

It should be noted that a total number of 500 epochs are considered for the training of the desired network.

**4.5. Simulation Results**

In this section, simulation results are presented. All the simulations were carried out on a computer with a Core i9 CPU, 1080 Titan GPU, and 64 GBs of RAM at the High-Performance Computing Center (HPCC) of the Ferdowsi University of Mashhad.

To present simulation results, we first show the classification results using an unoptimized CNN. In Figure 10, the resulting confusion matrix for the classification of 7 and 15 classes for the two datasets is demonstrated. It should be noted that to provide better evaluation, the values in these tables are normalized and presented in percentage.

87.5	12.5	0	0	0	0	0
0	100	0	0	0	0	0
0	0	100	0	0	0	0
0	0	0	83.3	0	16.6	0
0	0	0	0	100	0	0
0	0	0	16.6	0	83.3	0
0	0	0	0	0	0	100

100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	93.3	0	3.3	0	0	0	0	0	0	3.3	0	0	0	0
0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
0	0	0	3.3	0	96.6	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	96.6	0	0	3.3	0	0	0	0	0
0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure 10. The confusion matrix for using only CNN before the optimization of hyperparameters. Top: QMUL dataset, Down: T15 dataset.

Next, the hyperparameters of the deep CNN are optimized using WOA. The resulting confusion matrix for the classification using optimized DCNN is as Figure 11.

By comparing the results in Figure 10 and Figure 11, the profitable effect of the optimization of hyperparameters in improving the results is evident.

In the next step, we examine the effect of using only the ANFIS-based classifier on the test dataset. The results for the classification can be seen in Figure 12 in the form of the confusion matrix.

87.5	12.5	0	0	0	0	0
0	100	0	0	0	0	0
0	0	100	0	0	0	0
0	0	0	100	0	0	0
0	0	0	0	100	0	0
0	0	0	16.6	0	83.3	0
0	0	0	0	0	0	100

100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	96.6	0	0	0	0	0	0	0	3.3	0	0	0	0	0
0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	96.6	0	0	3.3	0	0	0	0	0	0
0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	3.3	100	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
0	0	0	0	0	0	0	0	0	3.3	0	0	0	96.6	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure 11. The confusion matrix for the implementation using only CNN after the optimization of hyperparameters. Top: QMUL dataset, Down: T15 dataset

87.5	0	0	0	0	12.5	0
0	100	0	0	0	0	0
0	0	83.3	16.6	0	0	0
0	0	0	100	0	0	0
0	0	0	0	100	0	0
0	0	0	0	16.6	83.3	0
0	16.6	0	0	0	0	100

100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.3	96.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	96.6	3.3	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	96.6	3.3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	96.6	0	0	0	0	0	3.3	0	0
0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure 12. The confusion matrix for using only ANFIS. Top: QMUL dataset, Down: T15 dataset

Due to the appropriate diversity of these two classifiers, the classification process using a structure combined from both networks is expected to offer excellent results. Hence, we combine the most appropriate CNN resulting from the optimization of the hyperparameters with the ANFIS network to achieve better results. In Figure 13, the results for the combination of these two classifiers are demonstrated.

The results demonstrated in Figure 13 clearly indicate the validity of the efficiency of the proposed hypothesis of combining the optimized CNN with the ANFIS algorithm.

Following the process of our simulation, we now provide

the results for the training of the autoencoder network.

First, we present the histogram for the final reconstruction error function related to the first dataset, using the first autoencoder network, and for the first class. Figure 14 illustrates this histogram.

The accumulation of the reconstruction error histograms for the seven networks can be seen in Figure 15.

87.5	12.5	0	0	0	0	0
0	100	0	0	0	0	0
0	0	100	0	0	0	0
0	0	0	100	0	0	0
0	0	0	0	100	0	0
0	0	0	0	0	100	0
0	0	0	0	0	0	100

100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	3.3	96.6	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	3.3	96.6	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0

Figure 13. The confusion matrix for the combination of CNN and ANFIS. Top: QMUL dataset, Down: T15 dataset

It should be noted that the above histograms are for normal data. For abnormal data, this histogram forms another shape shown in Figure 16.

As is evident in the above figures, each class in the first dataset has a specific range for the reconstruction error values. For normal and abnormal data, these ranges overlap. The utilization of deep CNNs and ANFIS networks aims to find a specified range and overcome this overlap, which provided excellent results, as shown in Table 2. It should be noted that these results are yielded through 10-fold cross-validation criteria. In other words, they are the average value of ten simulations, where the data is folded into 10 parts, and one part is taken for test and others for training. This procedure is repeated for all parts.

Based on the results mentioned above, it is clear that our proposed method is capable of detecting anomalies –in addition to normal incidents– with a remarkable accuracy compared to other successful updated methods in anomaly detection. Our method offers approximately 10 percent improvement compared to the state-of-the-art studies in the

QMUL dataset and 0.2 percent in the T15 dataset. In other words, the simulation results indicate the superior performance of the proposed method and prove its accountability in the anomaly detection process. This superior performance is attained through using several efficient factors of fuzzy flexibility, CNN capability, and autoencoder susceptibility in anomaly patterns.

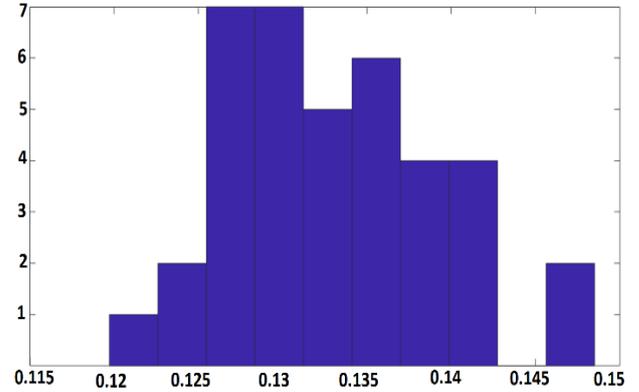


Figure 14. Histogram for the reconstruction error related to the first class in the QMUL dataset

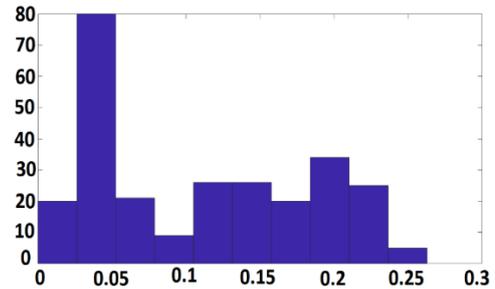


Figure 15. The accumulation of the reconstruction error histograms for the seven autoencoder networks

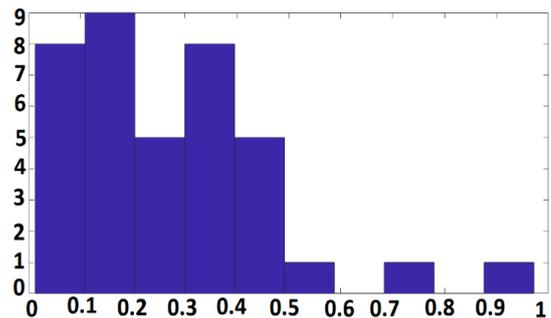


Figure 16. The reconstruction error histogram for the abnormal data in the QMUL dataset

Table 2. Accuracy percentage in the detection of anomalies versus methods proposed in other studies

Dataset	Proposed Method	MLL [27]	A-HDBSCAN [26]	DPMM [25]	TUIC [24]	BP [23]
QMUL	87.5	78.08	74.66	75.34	78.08	73.97
T15	99.5	99.3	97.87	98.26	98.07	88.93

## 5. Conclusion

In this study, a novel method for anomaly detection in the two datasets of QMUL and T15 is proposed. This method is based on the utilization of two classifiers: optimized deep Convolutional Neural Network (ODCNN) and Adaptive Neuro-Fuzzy Inference System (ANFIS), along with an autoencoder network. One innovation of this study is employing Whale Optimization Algorithm (WOA) to achieve a proper structure for the CNN by determining the best values for its hyperparameters, including initial learning rate, momentum rate, and regularization norm. The other innovation is the use of autoencoder networks to obtain an optimized structure for the anomaly detection process. As demonstrated in the simulations, the proposed optimization process yields an efficient structure. Achieving an accuracy percentage of 87.5 and 99.5 for the QMUL and T15 datasets, respectively, shows its superiority in comparison to other studies and indicates its proper performance.

## References

- [1] Z. Karimi Zandian, and M. R. Keyvanpour, "SSLBM: A New Fraud Detection Method Based on Semi-Supervised Learning," *Computer and Knowledge Engineering*, vol. 2, no. 2, pp.10-18, 2020.
- [2] N. Ab azar, A. Shahmansoorian, M. Davoudi, "Uncertainty-aware Path Planning Using Reinforcement Learning and Deep Learning Methods," *Computer and Knowledge Engineering*, 2020.
- [3] S. Kianian, S. Farzi, "Assessment of Customer Credit Risk using an Adaptive Neuro-Fuzzy System," *Computer and Knowledge Engineering*, vol. 2, no. 2, pp.19-28, 2020.
- [4] S. Kelathodi Kumaran, D. Prosad Dogra, P. Pratim Roy, A. Mitra, "Video Trajectory Classification and Anomaly Detection Using Hybrid CNN-VAE," *arXiv preprint arXiv:1812.07203*, 2018.
- [5] S. Mirjalili, A. Lewis, "The whale optimization algorithm. *Advances in engineering software*," 95, pp.51-67, 2016.
- [6] E. Barucija, A. Mujcinovic, B. Muhovic, E. Zunic, D. Donko, "Data-driven approach for anomaly detection of real GPS trajectory data," *In2019 XXVII International Conference on Information, Communication and Automation Technologies (ICAT) IEEE*, pp. 1-6, 2019.
- [7] MY. Choong, L. Angeline, RK. Chin, KB. Yeo, KT. Teo. "Modeling of vehicle trajectory using K-means and fuzzy C-means clustering," *IEEE International Conference on Artificial Intelligence in Engineering and Technology (ICAET) IEEE*, pp. 1-6, 2018.
- [8] Y. Li, T. Guo, R. Xia, W. Xie, "Road traffic anomaly detection based on fuzzy theory," *IEEE Access*, vol. 6, pp.40281-8, 2018.
- [9] PA. Kumar, V. Vaidehi, "A transfer learning framework for traffic video using neuro-fuzzy approach," *Sādhanā*, vol. 42, no. 9, pp.1431-42, 2017.
- [10]R. Nawaratne, D. Alahakoon, D. De Silva, X. Yu, "Spatiotemporal anomaly detection using deep learning for real-time video surveillance," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 393-402, 2019.
- [11]M. Moallem, A. Pouyan, "Anomaly Detection using LSTM AutoEncoder," *Journal of Modeling in Engineering*, vol. 17, no.56, pp. 191-211, 2019.
- [12]A. Aboah, "A vision-based system for traffic anomaly detection using deep learning and decision trees," *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* , pp. 4207-4212, 2021.
- [13]X. Zhang, Y. Zheng, Z. Zhao, Y. Liu, M. Blumenstein, and J. Li, "Deep learning detection of anomalous patterns from bus trajectories for traffic insight analysis," *Knowledge-Based Systems*, 217, p.106833, 2021.
- [14]K. Rezaee, S.M. Rezakhani, M.R. Khosravi, and M.K. Moghimi, "A survey on deep learning-based real-time crowd anomaly detection for secure distributed video surveillance," *Personal and Ubiquitous Computing*, pp.1-17, 2021.
- [15]M.G. Narasimhan, and S. Kamath, "Dynamic video anomaly detection and localization using sparse denoising autoencoders," *Multimedia Tools and Applications*, vol.77, no.11 , pp.13173-13195, 2018.
- [16]J. Zhao, Z. Yi, S. Pan, Y. Zhao, Z. Zhao, F. Su, and B.Zhuang, "Unsupervised Traffic Anomaly Detection Using Trajectories," *In CVPR Workshops*, pp. 133-140, 2019.
- [17]HO. Ahmed, ML. Wong, AK. Nandi, "Intelligent condition monitoring method for bearing faults from highly compressed measurements using sparse over-complete features," *Mechanical Systems and Signal Processing*, 99 pp,459-77, 2018.
- [18]J.S. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE transactions on systems, man, and cybernetics*, vol 23, no 3, pp.665-685, 1993.
- [19]H. Zenati, CS. Foo, B. Lecouat, G. Manek, VR. Chandrasekhar. "Efficient gan-based anomaly detection," *arXiv preprint arXiv:1802.06222*, 2018.
- [20]C. C. Loy, T. Xiang, and S. Gong, "From local temporal correlation to global anomaly detection," *In ECCV*, 2008.
- [21]H. Xu, Y. Zhou, W. Lin, and H. Zha, "Unsupervised trajectory clustering via adaptive multi-kernel-based shrinkage," *In ICCV*, 2015.
- [22]DP. Kingma, M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*. 2013.
- [23]H. Averbuch-Elor, N. Bar, D. Cohen-Or, "Border-Peeling Clustering," *IEEE transactions on pattern analysis and machine intelligence*. Vol 42,no 7, pp.1791-7, 2019.
- [24]KK. Santhosh, DP. Dogra, PP. Roy, "Temporal unknown incremental clustering model for analysis of traffic surveillance videos," *IEEE Transactions on Intelligent Transportation Systems*, vol 20, no 5, pp. 1762-73, 2018.

- [25]DM. Blei, MI. Jordan, "Variational inference for Dirichlet process mixtures," *Bayesian analysis*, vol. 1, no. 1, pp.121-43, 2006.
- [26]L. McInnes, J. Healy, S. Astels, "hdbscan: Hierarchical density based clustering," *Journal of Open Source Software*, vol 2, no. 11, pp.205, 2017.
- [27]SK. Kumaran, A. Chakravarty, DP. Dogra, PP. Roy, "Likelihood learning in modified Dirichlet Process Mixture Model for video analysis," *Pattern Recognition Letters*, pp.128:211, 2019.