An Enhanced Spatiotemporal Deep Learning Model for Anomaly Detection in Vehicular Ad Hoc Networks and Smart Parking Systems

KHADIJA MOUATASSIM[1*], ABDELFETTAH MABROUK[1],
CHAIMAE OUCHICHA[2*]
[1]Department of Computer Science and Mathematics,
Chouaïb Doukkali University, ESTSB, ELITES
Lab, El Jadida,
MOROCCO

[2]Department of Management Assistance Techniques,
Hassan 1st University (UH1), ENCG, FAMISDS Lab, Settat,
MOROCCO

[*]*Corresponding Author*

Abstract: Smart parking and traffic management are some of the major problems faced during the development of a smart city. There exist numerous challenges when trying to develop systems to solve the above-mentioned problem since Vehicular Ad-Hoc Networks (VANETs) produce a large volume of spatiotemporal data (vehicle speed, direction, and GPS position) in real-time. Nevertheless, due to their complexity, it is difficult to detect the existence of any abnormal events rapidly and reliably. In order to tackle such challenges, we suggest developing a hybrid deep neural network model, integrating Long-Short Term Memory (LSTM) network capable of capturing long-term dependencies and Squeeze-and-Excitation block (attention mechanism) highlighting important features. The proposed model is tested on AV-GPS-Dataset, containing GPS tracks of real vehicles as well as attack instances. Experiments show excellent results (accuracy equals to 99.81%, precision to 99.81%, recall to 99.73%, and F1 score to 99.74%), which outperform traditional methods significantly. Our results confirm that spatiotemporal data plays a vital role in solving problems stated.

Key-Words: Vehicular Ad Hoc Networks (VANETs), Spatiotemporal Data, Anomaly Detection, Deep Learning, LSTM, Squeeze-and-Excitation, Smart Parking.

Received: November 24, 2025. Revised: January 19, 2026. Accepted: February 21, 2026. Published: June 5, 2026.

1 Introduction

Today, most cities employ technology and form a connected and intelligent transportation system using VANETs (Vehicular Ad-hoc Networks), [1]. Such a system increases road safety, improves the efficiency of traffic management, and allows the transfer of critical information with drivers. Nevertheless, the employment of Internet technologies in the field creates numerous challenges concerning security issues. As vehicles become more interconnected, they face numerous cybersecurity risks that threaten not only the vehicles but also the lives of passengers. What is more, such systems may be attacked by hackers trying to violate rules and create anomalies in urban infrastructure, [2]. For instance, a vehicle driving at a wrong place and in the opposite direction may affect road safety drastically. Data collected in such a case, like vehicle speed and GPS location, is rather complex and difficult to analyze when a certain event takes place. The Intrusion Detection System (IDS) becomes a real challenge when dealing with such cases, [3]. It is precisely machine learning models that help solve this problem. With the help of them, we can easily distinguish an attack event from a normal one. To overcome the abovementioned challenge, a new hybrid deep neural network model ST-ADNet will be developed. Its main advantage lies in the ability to process a large volume of complex data and make accurate predictions concerning a particular event. Also, it should be stressed that the proposed algorithm is able to examine both temporal data (speed, direction, etc.) and spatial data (GPS position). This helps gain deeper insight into vehicle behavior and improve anomaly detection in such a way. ST-ADNet implies two improvements compared to other models. First, it uses Squeeze-and-Excitation (SE) block, [4] which works on the basis of the attention mechanism, assigning importance to data features. It means that any important event gets more attention, for instance, a quick change in speed. Such an approach makes the algorithm more powerful and efficient. The second improvement deals with spatiotemporal data fusion. Combining these two kinds of data allows us to get a better idea of a vehicle at a particular point in time. For experimental evaluation of the suggested approach, we use AV-GPS-Dataset, which contains autonomous vehicle GPS data, as well as GPS spoofing attack instances in the real world. The data provided in this database are rather realistic and can be employed for testing IDS algorithms.

This paper is structured as follows. The related work is covered in section 2. The methodology, data, and proposed model are described in section 3. Section 4 contains metrics. The results of experiment and their discussion are shown in section 5. Conclusion is provided in section 6.

2 RELATED WORK

There have been numerous studies in the sphere of anomalous detection in autonomous vehicle systems and smart parking that employ spatio-temporal databases. Various researchers employ these datasets for anomaly detection and attack identification on vehicles and parking systems. The authors in, [5] proposed a framework for GPS spoofing attacks' detection in autonomous vehicles. The framework uses AV-GPS data in order to detect abnormal behaviors in the movement of vehicles. It compares observed events to the expected trajectory calculated via physics-based and machine learning-based approaches to reveal abnormalities in the movement. Such an approach shows quite good performance with the accuracy of 97.2%. The authors in, [6] suggested an anomaly detection system for connected vehicles using learning from demonstrations (LfD) technique. The system employs a set of expert demonstrations to learn about the proper trajectory of a vehicle. The system identifies abnormal behavior by comparing the trajectories observed in reality and the trajectory predicted by the learned policy. The performance of the approach was evaluated via accuracy, equal to 94%. The authors in, [7] developed a new advanced anomaly detection framework for autonomous vehicles. The framework employs a Modified Convolutional Neural Network (M-CNN). By fusing spatiotemporal data with sensor data, the model detects abnormalities in the movement of vehicles in real time. Due to the deep learning method, the accuracy of the framework reaches 99.4%. The authors in, [8] proposed an ensemble learning framework for anomaly detection in autonomous driving. The study compares the performance of different ensemble learning methods employing two real-world datasets and determines their effectiveness in the task of anomaly detection. In this case, the accuracy reaches 80.0%. The authors in, [9] proposed a robust anomaly detection method for GPS spoofing attacks on intelligent autonomous vehicles. The technique uses statistical and machine learning-based approaches to distinguish falsified from authentic data. The PDOP, HDOP, and VDOP were used to identify spoofing. Among the features mentioned PDOP proved to be the best one providing accuracy of 95.8%.

3 The proposed model

In this section, we introduce a novel model that uses deep learning techniques to detect GPS spoofing in autonomous vehicles, [10]. As described above, the model uses both spatial and temporal aspects of the data obtained from GPS devices, [11]. It uses convolutional neural networks, bidirectional LSTMs, [12] and squeeze-and-excitation blocks, [13] to extract meaningful features from the dataset. Detailed architecture of the model, its preprocessing procedures, training techniques, and evaluation metrics will be presented in the subsections that follow. Fig. 1 (Appendix), Table 1 (Appendix) and Table 2 (Appendix) represent the structure of our model.

The proposed spatio-temporal deep learning model utilizes the temporal and spatial characteristics of GPS data for intrusion detection. Its architecture comprises two independent branches designed for heterogeneous data, using CNN, RNN layers, and squeeze-and-excitation. The features extracted from the data include temporal variables such as vehicle velocity, direction, and time steps as well as spatial features extracted from positioning and parking coordinates. All features will undergo the standardization procedure for normalization, after which they will be reshaped to prepare them for sequence-based learning. Each branch processes a distinct group of features, with convolutional layers extracting local features, followed by batch normalization and dropout. Afterward, bidirectional LSTM layers will extract temporal relations in the forward and backward directions. To refine the feature representation, we introduce a squeeze-and-excitation network that recalibrates the channels' importance by highlighting the most relevant features and ignoring irrelevant ones. The representations will be concatenated and regularized with the help of dropout, followed by another round of convolution to extract features on a deeper level. This output is then flattened and passed to fully connected layers for classification. Fully connected layers yield binary classification outputs of intrusions, but the side autoencoder part produces anomaly data. This deep learning network model is optimized using the Adam optimizer and the categorical cross-entropy as a loss function.

3.1 DATASET Description: AV-GPS-Dataset

The AV-GPS-Dataset, [14] was initially created with the aim of facilitating studies in the area of autonomous driving vehicles, while at the same time being well-suited for use in VANETs. The accuracy offered by the GPS sensor data allows us to simulate vehicle trajectory, increase localization accuracy due to the fusion of GPS and V2V data, improve inter-vehicle communication and cooperation, and assess the effectiveness of the proposed protocols for VANET applications, such as cooperative trajectory planning. The dataset consists of two major categories of AV-GPS-Dataset – AV-GPS-Dataset (Real Data) and AV-GPS-Dataset

(Simulation-Generated Data), each consisting of three subsets in total (thus there are six subsets overall): subsets 1–3 form AV-GPS-Dataset (Real Data), while subsets 4–6 belong to AV-GPS-Dataset (Simulation-Generated Data). Each subset has two classes of data – Normal and Attack data, all of which are extracted in CSV format. The AV-GPS-Dataset focuses on collecting temporal features of autonomous vehicles (AVs) by using Autonomous Vehicle Testbed (AVT) and CARLA simulator environments. In this research, we will use the AV-GPS-Dataset-1, the first subset of the broader AV-GPS-Dataset, one of the many datasets collected in the course of our research on autonomous vehicles (AVs). The first subset is the largest in the dataset, comprising $62,042$ data samples in total, out of which approximately 75% were labeled as normal $(46,787)$ and 25% as attacks $(15,255)$. Acquisition of data samples was performed in two locations on the campus of University of Arizona specifically chosen to provide maximum GPS signal availability and minimal environment noise. Separate sessions were made for collecting normal and attacked data samples, after which they were combined into the dataset. It is available in the CSV format, with each sample having temporal features of the vehicle dynamics, which include speed, heading, and geospatial coordinates obtained from GPS position of the vehicle.

In order to make the dataset analysis more extensive, some extra parameters were added to facilitate the simulation of AV motion and improve anomaly detection. One of the most significant additions to the dataset is the inclusion of parking lot coordinates. The centroid of each parking lot was computed by taking the average of the latitude and longitude values of all points that define it. Mathematically, for a parking zone composed of n coordinate points $(lat_i, long_i)$, the centroid coordinates are calculated using Eq. 1 and Eq. 2:

$$AverageLatitude = \frac{1}{n}\sum_{i=1}^{n} lat_i \tag{1}$$

$$AverageLongitude = \frac{1}{n}\sum_{i=1}^{n} long_i \tag{2}$$

The calculated coordinates were incorporated into the dataset under the columns Parking Latitude and Parking Longitude, enabling more accurate modeling of vehicle movement within parking areas, which are zones characterized by expected behaviors such as stopping or slowing down. This addition not only improves trajectory analysis but also increases the dataset's ability to identify anomalous vehicle behavior.

In addition, parking areas function as fixed geospatial reference points that support the validation of vehicle trajectories and the detection of irregular movement patterns. Their integration into the AV-GPS-Dataset-1 improves the precision of navigational data and strengthens the system's capability to detect GPS spoofing or trajectory manipulation. Consequently, this enhancement contributes to the development of more secure, reliable, and resilient autonomous navigation systems capable of maintaining operational integrity and resisting external interference.

3.2 Data Preparation

The dataset $\mathcal{D}$ consists of samples X, which are divided into temporal features X_t and spatial features X_s, such that $X = X_t, X_s$. To ensure consistent input scaling, normalization is applied to each subset of the data. This step prevents the model from being biased by differences in feature ranges during training, as shown in Eq. 3 and Eq.4.

$$\hat{X}_t = \frac{X_t - \mu_t}{\sigma_t} \tag{3}$$

$$\hat{X}_s = \frac{X_s - \mu_s}{\sigma_s} \tag{4}$$

Here, μ and σ represent the mean and standard deviation of each feature set, respectively. In addition, categorical variables are processed using One-Hot Encoding, which converts each category into a binary vector representation, as shown in Eq. 5.

$$\text{OneHot}(c) = [0, \ldots, 1, \ldots, 0] \tag{5}$$

In this representation, the position of the value '1' corresponds to the index of the category. This preprocessing step is important because it converts the data into a form suitable for neural network training. Normalization helps prevent features with larger numerical ranges from dominating the learning process, which could otherwise slow or destabilize convergence. One-Hot Encoding, on the other hand, ensures that categorical variables are represented without introducing artificial ordinal relationships, thereby preserving their original semantic meaning.

3.3 Feature Extraction from Temporal and Spatial Data

The temporal and spatial inputs are processed through parallel branches, each built with convolutional and recurrent layers to learn meaningful patterns specific to their respective domains.

3.4 Convolutional Layer

For each position i in the sequence, the convolution operation is defined as in Eq. 6.

$$y_i = \text{ReLU}\left(\sum_{j=0}^{k_{\text{size}}-1} W_j \cdot x_{i+j} + b\right) \tag{6}$$

Here, $k_{size} = 3$ represents the kernel size, and W_j are the learnable convolution filters. This operation captures local temporal and spatial dependencies by sliding the kernel across the input sequence. Batch Normalization is then applied after the convolution step to stabilize and normalize the outputs, which helps accelerate training and improves overall stability by mitigating internal covariate shift, as shown in Eq. 7:

$$\text{BN}(z) = \gamma \frac{z - \mathbb{E}[z]}{\sqrt{\text{var}(z) + \epsilon}} + \beta \tag{7}$$

where ϵ is a small constant added for numerical stability, and the learnable parameters are γ (scale) and β (shift).

3.5 L_2 regularization

L_2 regularization is incorporated into the loss function to penalize large weight values, helping to reduce overfitting, as defined in Eq. 8. This encourages the model to learn smoother and more generalizable representations by discouraging excessively large parameter magnitudes.

$$L_{\text{reg}} = \lambda \sum_{w \in W} w^2 \tag{8}$$

3.6 Dropout

During training, dropout with a rate $p = 0.7$ is applied by randomly deactivating neurons, as described in Eq. 9. This regularization strategy improves generalization by forcing the network to learn redundant and more robust feature representations.

$$\hat{y}_i = y_i \cdot m_i \quad , \quad m_i \sim \text{Bernoulli}(1-p) \tag{9}$$

3.7 Bidirectional LSTM (BiLSTM)

The BiLSTM module is intended to learn the dependencies that exist in the data in the forward and backward time directions. The BiLSTM module employs an internal gate mechanism for controlling the flow of information and maintaining memory (Eqs. 10, 11, 12, 13, 14, and 15).

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{10}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{11}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{12}$$

$$\tilde{c}t = \tanh(W_c x_t + U_c ht - 1 + b_c) \tag{13}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{14}$$

$$h_t = o_t \odot \tanh(c_t) \tag{15}$$

This architecture is particularly effective for sequential data, as it captures temporal dependencies in a structured and bidirectional manner.

3.8 Squeeze and Excite (SE) Block

In order to improve the feature representation, an SE block is included in each of the branches. This is done by first applying global average pooling over the temporal dimension using Eq. 16 below.

$$z_c = \frac{1}{T} \sum_{t=1}^{T} u_{c,t} \tag{16}$$

This operation compresses each channel c into a single scalar value, effectively capturing global contextual information. Subsequently, two fully connected layers are used to learn inter-channel dependencies, as shown in Eq. 17.

$$s = \sigma\big(W_2, \text{ReLU}(W_1 z)\big) \tag{17}$$

where σ represents the sigmoid activation function. The resulting scaling coefficients s_c are then used to recalibrate each channel, as expressed in Eq. 18, enhancing informative channels while suppressing less relevant ones.

$$\widetilde{u}c, t = s_c.uc, t \tag{18}$$

Convolutional layers are responsible for extracting short-term patterns, whereas the BiLSTM captures long-range temporal dependencies, which is crucial for complex time-series and spatial data. Batch Normalization and dropout further stabilize training and help reduce overfitting. In this context, the SE block functions similarly to a self-attention mechanism by adaptively assigning importance to feature channels.

After obtaining refined temporal features $\widetilde{u}t$ and spatial features $\widetilde{u}s$ from their respective SE blocks, they are concatenated as shown in Eq. 19:

$$F = [\widetilde{u}t, \widetilde{u}s] \tag{19}$$

A dropout layer with a rate of 0.5 is then applied to the concatenated vector F to reduce overfitting, as given in Eq. 20.

$$F' = F.m \quad , \quad m \sim Bernoulli(0.5) \tag{20}$$

Finally, a deeper Conv1D layer with 512 filters and kernel size 3 is applied to the fused features, as shown in Eq. 21, followed by an additional dropout of 0.4.

$$y = Relu(Conv1D(F'))` \tag{21}$$

The separation of temporal and spatial feature extraction enables specialized learning for each modality, while their fusion allows the model to capture complex cross-domain interactions, resulting in richer representations and improved discriminative performance.

3.9 Classification Layer

The fused features are first flattened and a dropout layer with a probability of 0.7 to help minimize overfitting before making any predictions. This is followed by a fully connected layer with a softmax function to compute the output classes, as illustrated in Eq. 22.

$$\hat{y} = softmax(W_{out}.x + b_{out}) \tag{22}$$

In this equation, W_{out} represents the output layer weight matrix, b_{out} is the bias term, and x denotes the input feature vector. The softmax function is defined in Eq. 23.

$$softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{23}$$

Here, $\hat{y}$ represents the probability distribution over the two classes, Normal and Attack. The softmax layer produces interpretable probability scores that can be ranked or thresholded for decision-making. The strong dropout applied before this stage improves generalization and reduces the risk of overfitting on the training data.

3.10 Training and Evaluation

Training of the model takes place through the minimization of categorical cross-entropy loss, together with L_2 regularization, as seen in Eq. 24.

$$\mathcal{L} = -\sum_{i} y_i log\hat{y}_i + \lambda \sum_{w \in W} w^2 \tag{24}$$

In this case, y_i is the true class labels, while $\hat{y}_i$ stands for the probability predictions. Once the model has been trained, it is tested with a different dataset using performance measures such as accuracy, precision, recall, and the F1 score, which help determine how well it distinguishes between normal and attack behaviors.

4 Experimental Setup and Metrics

All simulations have been carried out using the Keras 3 framework on Python version 3.11.7. The whole implementation process took place using the computer that had a Windows 10 OS with Intel(R) Core(TM) i7-8850H CPU @ 2.60GHz and 8 GB RAM.

In order to train the model and test its performance, AV-GPS dataset was used. The said dataset was split randomly in two groups: one for training and validation (80% of the total), while the other one for testing purposes (20% of the total). Training was conducted during 50 epochs with the batch size equal to 64.

For model optimization, the Adam optimizer was used with the following parameters: learning rate $\eta = 0.0005$, coefficient for the decaying average of the gradients $\beta_1 = 0.9$, and coefficient for the decaying average of the squared gradients $\beta_2 = 0.999$. Activation functions for all hidden layers were ReLU.

To ensure generalization and avoid overfitting, early stopping technique with a patience of five epochs was used together with learning rate reduction on plateau with a reduction factor of 0.2 and minimum learning rate of 10^{-6}. For further information about the experimental environment, see Table 3 (Appendix). The performance of the proposed architecture is analyzed through several classification measures mentioned below:

- **The Accuracy** provides the percentage of correctly classified instances and can be determined by dividing the sum of correctly classified positive and negative instances by the total number of instances, as described in Eq. 25.

$$AC = \frac{TP + TN}{TP + TN + FP + FN} \tag{25}$$

  TP stands for true positives, which indicate the positive cases that were rightly predicted, whereas TN is true negatives and indicates the correctly predicted negative cases. On the other hand, FP means false positives, which are negative cases wrongly predicted as positives, and FN are false negatives, which are positive cases wrongly predicted as negatives.

- **The Precision** refers to the ratio of positive instances correctly identified by the classifier among all instances classified as positive. High precision indicates low number of false alarms, which is essential for anomaly detection systems. It is defined in Eq. 26 as:

$$Precision = \frac{TP}{TP + FP} \tag{26}$$

- **The Recall**, can be defined as the Sensitivity or the True Positive Rate, and shows how well the system is able to identify all the positives. High Recall implies efficient identification of positive instances such as attacks in security-related problems. It is calculated in Eq. 27 as:

$$Recall = \frac{TP}{TP + FN} \tag{27}$$

- **The F1-Score** is defined as the harmonic mean of precision and recall and provides a balanced approach to evaluating both parameters. It can be represented as shown in Eq. 28. below:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{28}$$

- **The AUC** represents Area Under the Curve and it is computed based on ROC. Values close to 1.0 indicate higher ability to discriminate between classes.

5 Experimental results and analysis

The results of this experiment show a thorough investigation into the proposed algorithm for anomaly detection using experiments carried out on the AV-GPS dataset. The evaluation of the analysis consists of both quantitative and comparative analysis. For starters, the effectiveness of the proposed algorithm is evaluated against standard performance measurements such as accuracy, precision, recall, and F1-score. Thereafter, a comparison is made between the performance of the proposed algorithm and other algorithms in the current literature.

5.1 Results and analysis

This section will discuss the results achieved by our proposed ST-ADNet model. The results include the classification outcomes, evaluation metrics, and the overall training behavior. Visualization tools such as the confusion matrix, ROC curve, and training plots are provided to illustrate the model's performance and its ability to distinguish between normal traffic and spoofing attacks in vehicular networks, [15] .

In Fig. 2 (Appendix), the confusion matrix of the proposed ST-ADNet anomaly detection model is presented. As observed, the model correctly classifies 9242 normal instances, while 11 normal instances are misclassified as spoofing attacks. For spoofing attacks, 3143 instances are correctly detected, whereas only 13 are incorrectly labeled as normal. These results indicate that the model achieves highly reliable discrimination between normal vehicle behavior and spoofing attacks, with very few false positives and false negatives. Such performance is essential for robust intrusion detection in vehicular networks, where both missed detections and false alarms must be minimized.

Table 4 (Appendix) presents the performance of the proposed ST-ADNet model for each individual class. As for the normal class, the model demonstrates an outstanding performance in terms of precision, recall, and F1-score, with the results being 99.86%, 99.88%, and 99.87% respectively. Thus, the presented findings confirm the high ability of the model to classify normal behavior without making errors. As far as the spoofing attack class is concerned, the precision, recall, and F1-score obtained are 99.76%, 99.59%, and 99.62%. Therefore, it can be argued that the model provides reliable results regarding detection of attacks in the given network. Overall, the accuracy, recall, and F1-score achieved by the model are 99.81%, 99.73%, and 99.74%, highlighting its robustness and reliability for anomaly detection in Vehicular Ad-hoc Networks.

Fig. 3 (Appendix) presents the progression of accuracy and loss for both training and validation sets over 22 epochs. From the first epoch, the model shows strong learning behavior, reaching a training accuracy of 92.02% and a validation accuracy of 97.57%. This initial difference suggests that the model already generalizes well even in the early stage of training.

As training continues, both accuracy curves rise quickly. The validation accuracy surpasses 99% by epoch 4, while the training accuracy gradually converges toward it. From around epoch 6 onward, the model becomes stable, with training accuracy settling near 98.9% and validation accuracy fluctuating between 99.5% and 99.8%. The close alignment of both curves indicates stable learning with no clear signs of overfitting, which is important for reliable real-world deployment.

At the same time, both training and validation loss decrease consistently, confirming effective optimization. The training loss reduces from 0.2480 to 0.0413, while the validation loss decreases from 0.0890 to 0.0161. The steadily declining validation loss further supports the model's strong generalization ability. Minor variations in the later epochs are expected and do not affect the overall trend. Overall, these results indicate that the proposed hybrid model learns efficiently and maintains strong generalization, making it suitable for anomaly detection in VANETs and smart parking systems.

As shown in Fig. 4 (Appendix), the ROC curve of the suggested model shows the connection between the True Positive Rate (sensitivity) and False Positive Rate at varying threshold levels. The plot stays close

to the top-left corner, meaning high performance, and moves towards the top right corner, suggesting high classification performance. The model achieves an AUC of 1.00, reflecting perfect separation between normal behavior and spoofing attacks. In practical terms, this indicates that the model achieves both high sensitivity and specificity, correctly identifying all spoofing attacks without producing false alarms.

5.2 Comparison with Other Methods

The progress in the research of the topic of anomaly detection within VANETs and smart parking systems has been significant lately owing to the development of spatio-temporal databases, such as AV-GPS data. In this paper, the comparative analysis of the performance of the proposed solution will be conducted with respect to several other state-of-the-art anomaly detection techniques.

As can be seen in Table 5 (Appendix), there are significant differences in terms of accuracy, precision, recall, and F1-scores between the ST-ADNet and some other anomaly detection methods. The GPS-IDS spoofing detection system (Model 1) has shown an accuracy of 97.2%, precision of 99.4%, recall of 89.9%, and F1-score of 94.4%. Model 2, that is Learning from Demonstrations, achieved an accuracy of 94%, precision of 92.86%, recall of 96.3%, and an F1-score of 94.55%. The best performance has been demonstrated by M-CNN architecture (Model 3) with accuracy of 99.4%, precision of 97.1%, recall of 98.7%, and an F1-score of 98.6%. Model 4, which is based on the concept of ensemble learning, has shown rather low performance rates (80.0% accuracy, 83% precision, 91% recall, and an F1-score of 86.0%). Machine learning-based model (Model 5) has resulted in accuracy of 95.8%, precision of 94.5%, recall of 93.2%, and an F1-score of 98%. As it can be seen from Table 5 (Appendix), the proposed model outperforms all the mentioned methods substantially. The accuracy rate of the proposed model (99.81%) is higher than Model 1 by 2.61%, than Model 2 by 5.81%, than Model 3 by 0.41%, than Model 4 by 19.81%, and than Model 5 by 4.01%. In terms of precision, the proposed solution has achieved 99.81% that is higher than Model 1 by 0.41% and higher than Model 2, Model 3, Model 4, and Model 5 by 6.95%, 2.71%, 16.81%, and 5.31%, correspondingly. In terms of recall, the proposed model gets a score of 99.73%, which outperforms Model 1 (9, 84% higher), Model 2 (3.43% higher), Model 3 (1.03% higher), Model 4 (8.73% higher), and Model 5 (6.53% higher). In terms of the F1-score, the proposed model has an accuracy score of 99.74%, which is 5.34% higher than that of Model 1, 5.19% higher than Model 2, 1.14% higher than Model 3, 13.74% higher than Model 4, and 1.74% higher than Model 5.

6 Conclution

In this paper, a novel spatiotemporal hybrid deep neural network architecture was suggested for anomaly detection in VANETs and smart parking infrastructure. Thanks to the integration of the LSTM layer and the Squeeze-and-Excitation mechanism, this model can successfully detect spatiotemporal relations and highlight the most important ones. On the basis of the performance tests performed using the AV-GPS-Dataset that includes real GPS trajectories and examples of spoofing attacks, the algorithm showed an outstanding accuracy of 0.9984. This performance exceeds all classical methods, proving the high reliability of spatiotemporal fusion for anomaly detection. The presented results are promising for the development of a new class of tools for securing cyber-physical systems. It is necessary to continue testing the proposed model using more extensive datasets in real-time conditions and increasing computational efficiency. Moreover, it is crucial to explore opportunities for detecting other types of cyber-physical attacks using the same approach.

References:

[1] Boukerche, A., Oliveira, H. A. B. F., Nakamura, E. F., Loureiro, A. A. F. Vehicular Ad Hoc Networks: A New Challenge for Localization-Based Systems. *Computer Communications*, 2008, vol. 31, no. 12, pp. 2838–2849. https://doi.org/10.1016/j.comcom.2007.12.004

[2] Diaz Ogas, M. G., Fabregat, R., and Aciar, S. Survey of smart parking systems. *Applied Sciences*, 2020, vol. 10, no. 11, pp. 3872. https://doi.org/10.3390/app10113872

[3] Al-Khulaidi, N. A., Zahary, A. T., Al-Shargabi, A. A., and Hazaa, M. A. S. Machine Learning for Intrusion Detection in Vehicular Ad-hoc Networks (VANETs): A Survey. In: *Proceedings of the 2024 4th International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, IEEE, 2024, pp. 1–10. https://doi.org/10.1109/eSmarTA62850.2024.10639016

[4] Hu, J., Shen, L., and Sun, G. Squeeze-and-Excitation Networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141. https://doi.org/10.1109/CVPR.2018.00745

[5] Abrar, M. M., Youssef, A., Islam, R., and Hassan, M. K. GPS-IDS: An Anomaly-based GPS Spoofing Attack Detection Framework for Autonomous Vehicles. *arXiv preprint*, arXiv:2405.08359, 2024. https://arxiv.org/abs/2405.08359 (Accessed: May 14, 2024)

[6] Yang, Z., Ying, J., Shen, J., Feng, Y., Chen, Q. A., Mao, Z. M., and Liu, H. X. Anomaly detection against GPS spoofing attacks on connected and autonomous vehicles using learning from demonstration. *IEEE Transactions on Intelligent Transportation Systems*, 2023, vol. 24, no. 9, pp. 9462–9475. https://doi.org/10.1109/TITS.2023.3269029

[7] Rajendar, S., and Kaliappan, V. K. Sensor Data Based Anomaly Detection in Autonomous Vehicles using Modified Convolutional Neural Network. *Intelligent Automation & Soft Computing*, 2022, vol. 32, no. 2, pp. 859–875. https://doi.org/10.32604/iasc.2022.020936

[8] Nazat, S., Alayed, W., Li, L., and Abdallah, M. Ensemble Learning Framework for Anomaly Detection in Autonomous Driving Systems. *Sensors*, 2025, vol. 25, no. 16, pp. 5105. https://doi.org/10.3390/s25165105

[9] Poornima, B., and Kumari, L. S. Detecting GPS spoofing in smart AVS: an accuracy-based machine learning approach. *International Journal of System Assurance Engineering and Management*, 2025, vol. 16, no. 2, pp. 581–594. https://doi.org/10.1007/s13198-024-02606-2

[10] Liu, S., Li, L., Tang, J., Wu, S., and Gaudiot, J.-L. *Creating Autonomous Vehicle Systems*. San Rafael, CA, USA: Morgan & Claypool Publishers, 2020. https://doi.org/10.2200/S01036ED1V01Y202007CSL012

[11] Yang, H., Zhang, X., Li, Z., and Cui, J. Region-level traffic prediction based on temporal multi-spatial dependence graph convolutional network from GPS data. *Remote Sensing*, 2022, vol. 14, no. 2, pp. 303. https://doi.org/10.3390/rs14020303

[12] Cheng, H., Xie, Z., Wu, L., Yu, Z., and Li, R. Data prediction model in wireless sensor networks based on bidirectional LSTM. *EURASIP Journal on Wireless Communications and Networking*, 2019, vol. 2019, no. 1, pp. 203. https://doi.org/10.1186/s13638-019-1511-4 format:

[13] Khan, S. Z., Mohsin, M., and Iqbal, W. On GPS spoofing of aerial platforms: A review of threats, challenges, methodologies, and future research directions. *PeerJ Computer Science*, 2021, vol. 7, article e507. https://doi.org/10.7717/peerj-cs.507

[14] Abrar, M. M. AV-GPS-Dataset: Autonomous Vehicle Global Positioning System. Dataset. GitHub, 2024. https://github.com/mehrab-abrar/AV-GPS-Dataset (Accessed: April 18, 2024)

[15] Van der Merwe, J., Zubizarreta, X., Lukčin, I., and Pérez, M. Classification of spoofing attack types. In: *Proceedings of the 2018 European Navigation Conference (ENC)*, IEEE, 2018, pp. 91–99. https://doi.org/10.1109/EURONAV.2018.8433227

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

Khadija Mouatassim and Chaimae Ouchicha contributed equally to the development of the model, as well as to the implementation, analysis, and interpretation of the results. Abdelfettah Mabrouk supervised the study from both scientific and methodological perspectives. All authors read and approved the final manuscript.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interests

The authors declare that no conflicts, financial or otherwise, existed during the conduct of this research

APPENDIX

Table 1: Architecture of the Proposed Deep Learning Model. Source: created by the authors.

Layer (Type)	Output Shape	Parameters	Connected To
InputLayer	(None, 1, 3)	0	–
InputLayer	(None, 1, 4)	0	–
Conv1D	(None, 1, 64)	640	Input (1,3)
Conv1D	(None, 1, 64)	832	Input (1,4)
Batch Normalization	(None, 1, 64)	256	Conv1D
Batch Normalization	(None, 1, 64)	256	Conv1D
Dropout	(None, 1, 64)	0	Batch Normalization
Dropout	(None, 1, 64)	0	Batch Normalization
Bidirectional LSTM	(None, 1, 128)	66,048	Dropout
Bidirectional LSTM	(None, 1, 128)	66,048	Dropout
Global Average Pooling 1D	(None, 128)	0	Bidirectional LSTM
Global Average Pooling 1D	(None, 128)	0	Bidirectional LSTM
Dense	(None, 2)	258	Global Avg Pooling
Dense	(None, 2)	258	Global Avg Pooling
Dense	(None, 128)	384	Dense
Dense	(None, 128)	384	Dense
Reshape	(None, 1, 128)	0	Dense
Reshape	(None, 1, 128)	0	Dense
Multiply (Gating)	(None, 1, 128)	0	BiLSTM + Reshape
Multiply (Gating)	(None, 1, 128)	0	BiLSTM + Reshape
Concatenate	(None, 1, 256)	0	Gated Features
Dropout	(None, 1, 256)	0	Concatenate
Conv1D	(None, 1, 512)	393,728	Dropout
Dropout	(None, 1, 512)	0	Conv1D
Flatten	(None, 512)	0	Dropout
Dense	(None, 64)	32,832	Flatten
Dropout	(None, 64)	0	Dense
Dense (Output Layer)	(None, 2)	130	Dropout

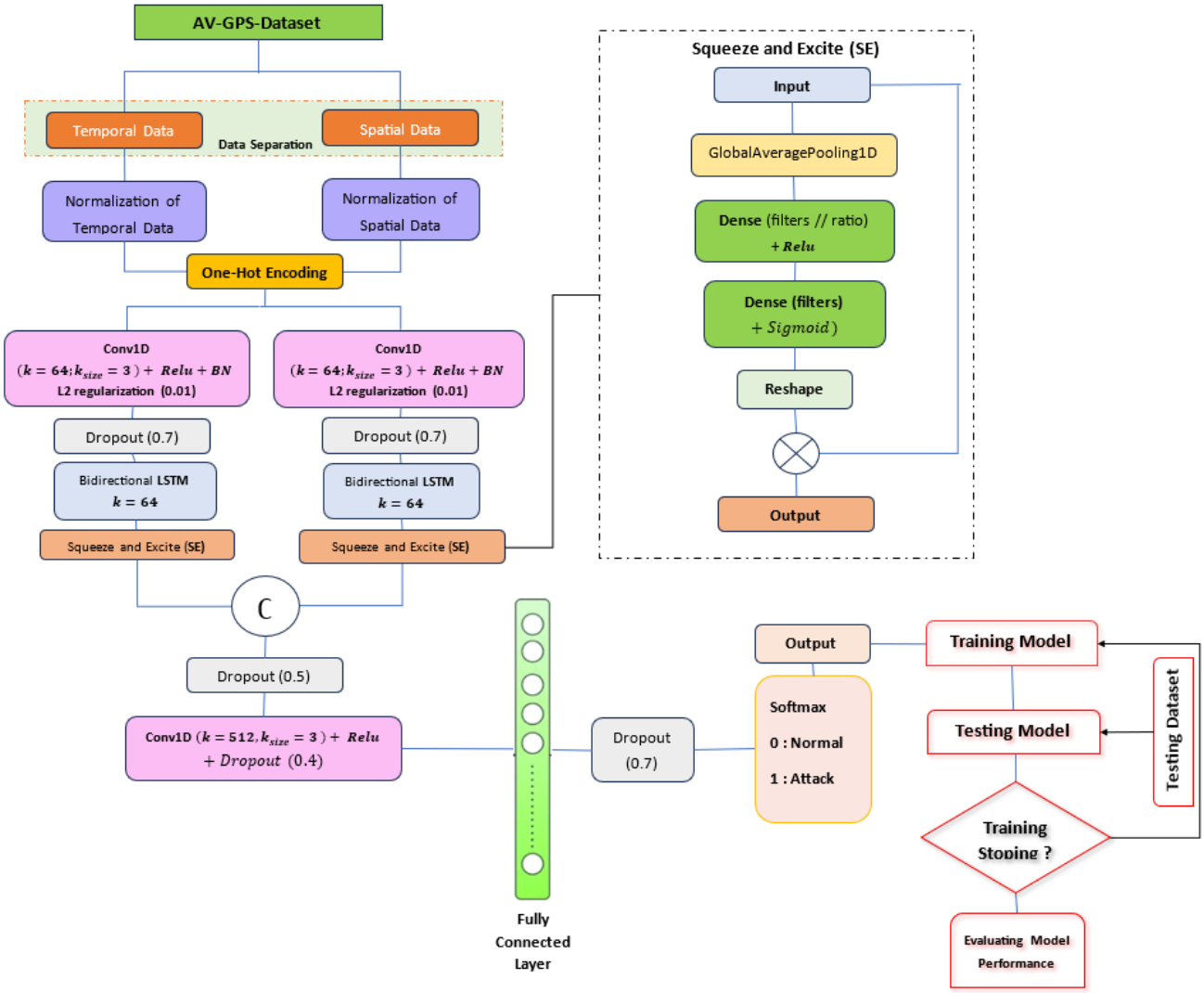

Figure 1: The structure of the proposed ST-ADNet model for IDS. Source: created by the authors.

Table 2: Compact Architecture of the Proposed Model. Source: created by the authors.

Layer (Type)	**Output Shape**	**Parameters**
InputLayer	(None, 1, 3)	0
InputLayer	(None, 1, 4)	0
Conv1D	(None, 1, 64)	640
Conv1D	(None, 1, 64)	832
Batch Normalization	(None, 1, 64)	256
Dropout	(None, 1, 64)	0
Bidirectional LSTM	(None, 1, 128)	66,048
Global Average Pooling 1D	(None, 128)	0
Dense	(None, 128)	384
Multiply (Gating)	(None, 1, 128)	0
Concatenate	(None, 1, 256)	0
Conv1D	(None, 1, 512)	393,728
Flatten	(None, 512)	0
Dense	(None, 64)	32,832
Dense (Output Layer)	(None, 2)	130

Table 3: The simulation system environment. Source: created by the authors.

project	**Environment/version**
Operating system	Windows 10 Professional
CPU	i7-8850H
memory	8Go
Keras	3.0.0
Development Environment	Python 3.11.7
learning rate	$\eta = 0.0005$
first moment decay rate	$\beta_1 = 0.9$
second moment decay rate	$\beta_2 = 0.999$

Table 4: Performance Metrics of ST-ADNet Model for Each Class. Source: created by the authors.

Class	**Precision**	**Recall**	**F1-Score**
Normal	0.9986	0.9988	0.9987
Spoofing attack	0.9976	0.9959	0.9962
Overall	0.9981	0.9973	0.9974

Table 5: Evaluation Metrics of ST-ADNet model compared to other models. Source: created by the authors.

Models	**References**	**Dataset**	**Accu-racy** (%)	**Preci-sion** (%)	**Recall (%)**	**F1-Score (%)**
Model 1	[5]	AVG-dataset 1	97.2	99.4	89.9	94.4
Model 2	[6]	Michigan Roundabout Dataset	94	92.86	96.30	94.55
Model 3	[7]	Safety Pilot Model Deployment (SPMD) dataset	99.4	97.1	98.7	98.6
Model 4	[8]	VeReMi Dataset	80	83	91	86
Model 5	[9]	GPS dataset	95.8	94.5	93.2	98
ST-ADNet	-	**AVG-dataset 1**	99.81	99.81	99.73	99.74

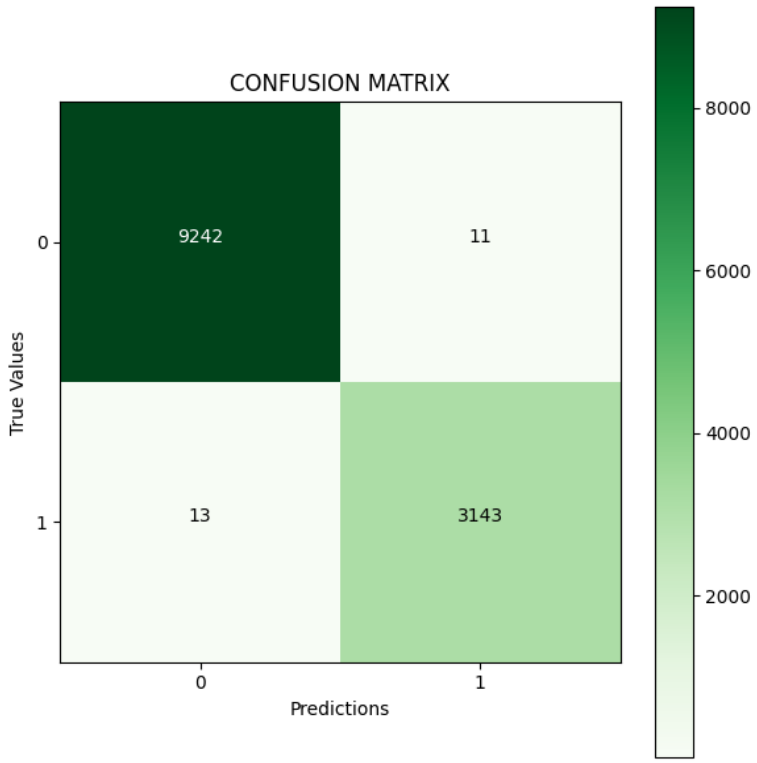

Figure 2: Confusion Matrix of ST-ADNet Model. Source: created by the authors.

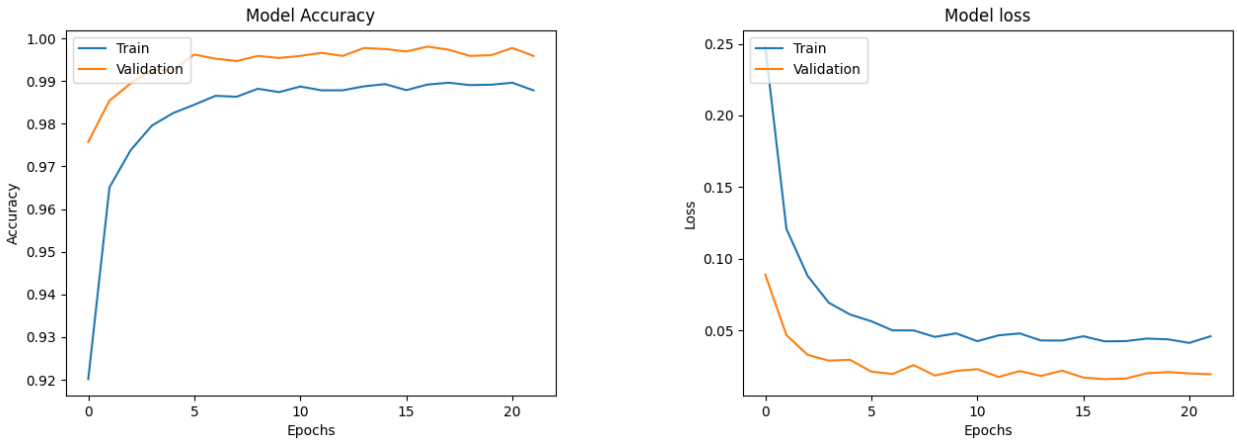

Figure 3: Training and Validation Accuracy and Loss over Epochs. Source: created by the authors.

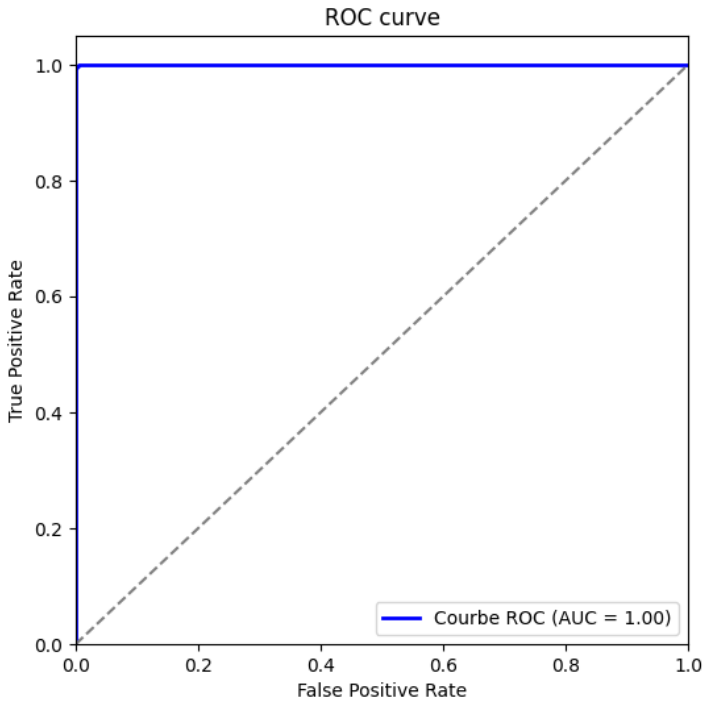

Figure 4: ROC Curve of ST-ADNet Model. Source: created by the authors.