scientific reports

Check for updates

OPEN

# CTDNN-Spoof: compact tiny deep learning architecture for detection and multi-label classification of GPS spoofing attacks in small UAVs

Ahmad Almadhor[1], Jamel Baili[2], Shtwai Alsubai[3], Abdullah Al Hejaili[4], Rastislav Kulhanek[5]✉ & Sidra Abbas[6]✉

**GPS spoofing presents a significant threat to small Unmanned Aerial Vehicles (UAVs) by manipulating navigation systems, potentially causing safety risks, privacy violations, and mission disruptions. Effective countermeasures include secure GPS signal authentication, anti-spoofing technologies, and continuous monitoring to detect and respond to such threats. Safeguarding small UAVs from GPS spoofing is crucial for their reliable operation in applications such as surveillance, agriculture, and environmental monitoring. In this paper, we propose a compact, tiny deep learning architecture named *CTDNN-Spoof* for detecting and multi-label classifying GPS spoofing attacks in small UAVs. The architecture utilizes a sequential neural network with 64 neurons in the input layer (ReLU activation), 32 neurons in the hidden layer (ReLU activation), and 4 neurons in the output layer (linear activation), optimized with the Adam optimizer. We use Mean Squared Error (MSE) loss for regression and accuracy for evaluation. First, early stopping with a patience of 10 epochs is implemented to improve training efficiency and restore the best weights. Furthermore, the model is also trained for 50 epochs, and its performance is assessed using a separate validation set. Additionally, we use two other models to compare with the *CTDNN-Spoof* in terms of complexity, loss, and accuracy. The proposed *CTDNN-Spoof* demonstrates varying accuracies across different labels, with the proposed architecture achieving the highest performance and promising time complexity. These results highlight the model's effectiveness in mitigating GPS spoofing threats in UAVs. This innovative approach provides a scalable, real-time solution to enhance UAV security, surpassing traditional methods in precision and adaptability.**

**Keywords** TinyML, Unmanned Aerial Vehicles (UAVs), Spoofing Attacks, Global Positioning Systems, Autonomous Vehicle, Machine Learning, Deep Neural Network

Unmanned Aerial Vehicles (UAVs), a particular kind of smart device, have become prevalent in current culture. These gadgets depend significantly on their communications structure[1,2], which is often built on the Internet of Things (IoT) networks and Global Positioning System (GPS) medium for every mission. GPS-based technologies suffer two primary dangers in UAVs: jamming and spoofing attacks[3]. The purpose of a jamming attack is denial-of-service (DoS), preventing the UAV from receiving the GPS signal. In a spoofing attack, the assailant duplicates and amplifies the GPS signal to serve the UAV's positional reference. The GPS and navigation system signal the correlation between higher power impacts. As a result, when the spoof signal is given to the UAV, it disregards the genuine GPS signal and begins to veer off track[4]. The target UAV cannot detect the drift during the attack since a correctly executed spoofing assault does not cause abrupt variations in the strength of the GPS signal received. Furthermore, the UAV cannot detect the drift because it is unaware of the proper position. These

[1]Department of Computer Engineering and Networks, College of Computer and Information Sciences, Jouf University, Sakaka 72388, Saudi Arabia. [2]Department of Computer Engineering, College of Computer Science, King Khalid University, Abha 61413, Saudi Arabia. [3]College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, AlKharj 16273, Saudi Arabia. [4]Faculty of Computers & Information Technology, Computer Science Department, University of Tabuk, Tabuk 71491, Saudi Arabia. [5]Department of Information Management and Business Systems, Faculty of Management, Comenius University in Bratislava, Odbojárov 10, 82005 Bratislava 25, Slovakia. [6]Department of Computer Science, COMSATS University, Islamabad, Pakistan. ✉email: rastislav.kulhanek@fm.uniba.sk; sidraabbas@ieee.org

factors make spoofing assaults difficult to identify[2]. Figure 1 presents the GPS spoofing attacks Scenario where it can be noticed that the UAV location can be shown somewhere else in the world.

In recent years, GPS service usage has increased significantly. By 2025, the market for GPS tracking devices is anticipated to grow to 3.38 billion from its present value of 1.57 billion[5]. In numerous instances, safety depends on tracking a moving object's location in the present moment. To help an automobile reach its destination, an autonomous vehicle's navigation system, for instance, receives GPS signals to determine the current latitude, longitude, acceleration, and direction. However, malevolent users or attackers have been encouraged to launch GPS attacks due to the rapid proliferation of GPS-enabled gadgets and affordable spoofing equipment. Because unencrypted GPS signals are so common and the ordinary GPS signal structure is accessible[6], it is simple for an assailant to launch a GPS spoofing assault from an externally programmable standard radio gadget like HackRF or USRP at a distance where the radio waves can interfere with the real GPS signals[7–10]. The target vehicle's navigation system will be duped into taking an incorrect path by the attacker once the attacker has control of the GPS signals in that area. Researchers have shown that while the vehicle is in autonomous mode, it is possible to use a HackRF to alter its trajectory or make it drive off-road [https://www.regulus.com/blog/tesla-model-3-spoofed-off-the-highway-regulus-research]. Aside from navigation, numerous additional applications and services have extensively used GPS data to enhance their offerings and user interfaces.

This research aims to identify and categorize GPS spoofing attacks in UAVs. This area has been explored through conventional machine learning methods such as Artificial Neural Networks (ANN)[11] and tree-based models[12], which provide efficient ways to detect these attacks. Ensemble learning techniques, such as Bagging and Boosting, have emerged as significant advancements in machine learning over the past decade, offering enhanced performance in cyber-attack detection[13]. Bagging techniques generate multiple datasets from the original data and continuously assess performance. At the same time, boosting adjusts the weight of observations based on the most recent classifications, leading to improved results over individual ML models[14]. However, these traditional models face challenges, including issues with output interpretation and bias, which can lead to problems with overfitting or underfitting. In response, we propose an innovative approach that integrates Self-Supervised Representation Learning (SSRL) with transfer learning techniques, enhancing model generalization and adaptability by leveraging unsupervised pre-training on large-scale data. This hybrid approach, combined with advanced deep learning models such as LSTM, GRU, and DNN architectures, significantly improves detection precision, surpassing the performance of conventional methods. This novel combination of SSRL and transfer learning not only addresses the limitations of traditional machine learning techniques but also paves the way for more robust, scalable, and accurate GPS spoofing detection systems in UAVs.

## Contribution

This paper makes the following contributions:

- Proposed a compact tiny deep learning architecture named *CTDNN-Spoof* for detection and multi-label classification of GPS Spoofing Attacks in Small UAVs.

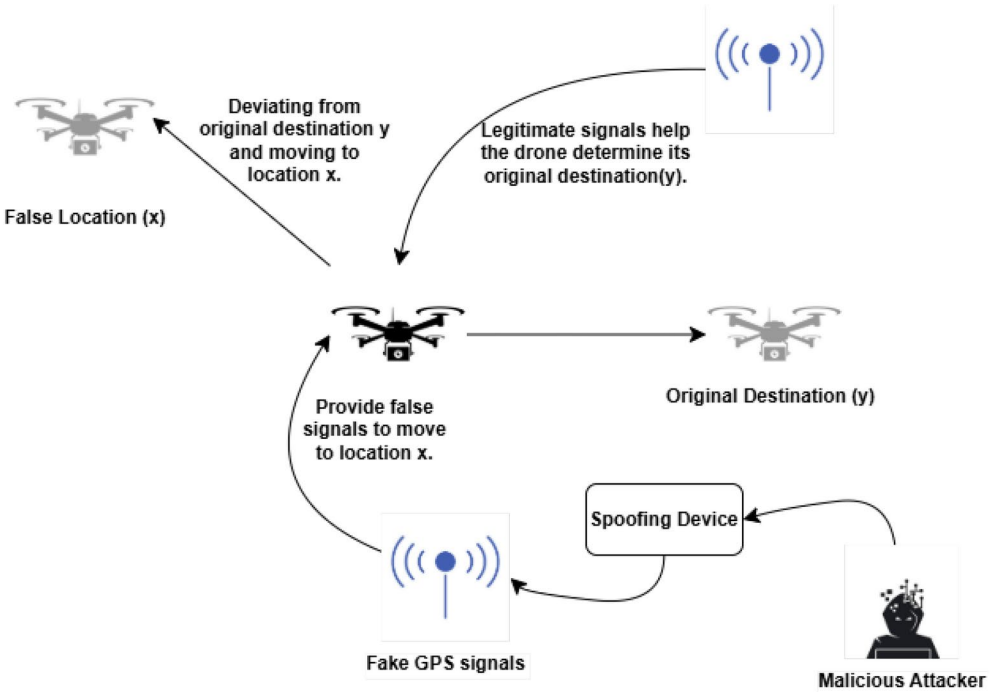


**Fig. 1**. Small UAV's GPS Spoofing Attacks Scenario.

- Furthermore, two other models are designed to compare with the proposed model in terms of complexity, loss, and accuracy. We implemented early stopping with the patience of 10 epochs and 50 epochs as well to evaluate the models' complexity and performance.
- The proposed *CTDNN-Spoof*, trained on various columns such as ch0_output, ch1_output, ch2_output, ch3_output, ch4_output, ch5_output, ch6_output, and ch7_output, exhibited varying but promising accuracies. The proposed model trained on the "ch6_output" column attained the highest accuracy, reaching a peak of 0.9912 during the 10 epoch and promising time complexity.

### Organization
The remainder of the article is organized as follows. Section 2 presents the relevant state-of-the-art research on GPS spoofing attacks in small UAVs. Section 3 presents the Compact Tiny Deep Learning Method for the Detection and Multi-Label Classification of GPS Spoofing Attacks in Small UAVs. Section 4 presents the experimental analysis, Results and Discussion. Finally, Section 7 concludes the paper.

## Related work
GPS spoofing attacks present a vulnerability to Unmanned Aerial Vehicles through which adversaries manipulate navigational signals to control UAVs illegally. Research endeavours about intrusion detection systems to counteract GPS spoofing attacks have become extensive because of mounting threat levels involving machine learning (ML) and deep learning (DL) methodologies.

### Machine learning and deep learning-based approaches
Various studies applied ML and DL models to detect and minimize GPS spoofing attacks. DeepPOSE serves as a deep learning-based solution that uses convolutional neural networks (CNNs) with recurrent neural networks (RNNs) for sensor noise filtering and real-time route reconstruction[15]. The proposed model performed effective trajectory corrections, which integrated sensor data with Google Maps projection features during detection while delivering high data recognition accuracy between datasets. They published their perception-data-based GPS spoofing detection method for UAVs in their article[16]. Real flight data combined with multiple ML classifiers reached detection rates higher than 99.69% using this method. The authors in[17] created two dynamic classifier selection methods known as Metric Optimized Dynamic Selector (MODS) and Weighted MODS to boost GPS spoofing detection performance. Through their single-stage ensemble feature selection method, they eliminated unimportant features, which resulted in a 99.6% accuracy rate with minimal wrong alarms. The anti-spoofing system developed by the authors of this paper[18] demonstrated superior performance through its 1D CNN design, which obtained 100% precision and 99% F1-score. The authors of this paper[19] trained a multi-layer perceptron (MLP) on statistical path loss data from base stations to reach above 93% detection accuracy when using three base stations. Authors of this paper[20] developed a joint system of deep neural networks for air traffic control identity authentication that includes zero-bias dense layers and continual learning capabilities. The framework worked successfully across different cyber-physical systems when applied to genuine ADS-B signals.

### Alternative approaches and hybrid techniques
Authors in[21] utilized various ML models for detecting spoofing attacks within ZigBee networks. Authors in[2] designed a Multi-layer Perceptron (MLP) model for UAV-based GPS spoofing detection, which yielded accuracy between 83.23% on the TEXBAT dataset and 99.93% on the MAVLINK dataset. Authors in[22] created a Lightweight, Trustworthy Message Exchange (LTME) scheme which linked trust management with cryptography to build reliable UAV network operations. The reputation update system of LTME, along with secret distribution methods, provided a dual mechanism for checking UAV identity validity and message authenticity. Authors in[23] developed the Intelligent Clustering Routing Approach (ICRA), which merged reinforcement learning-based clustering strategy adjustments with optimized routing for UAV Ad-hoc Networks (UANETs). This method improved both the network durability and minimised the end-to-end delay to demonstrate higher efficiency for clustering and energy consumption. PerDet, which represents a multi-sensor-based GPS spoofing detection system, achieved an extension from the author's original work in[24]. PerDet detected spoofing attacks with a 99.69% accuracy by merging data from accelerometers, gyros, magnets, GPS devices and barometers, which addressed sensor obstacles better than prior ML-based systems. Table 1 provides an overview of the major characteristics of GPS spoofing detection along with corresponding research methods.

### Key insights and research gap
Currently, available research shows that ML and DL-based solutions produce effective results for GPS spoofing detection. Many detection methods need custom adjustments for specific datasets, which complicates the process of achieving real-world generalization. The detection robustness of sensor fusion techniques alongside feature selection strategies needs additional optimization work. The proposed research implements an ensemble-based machine and deep learning framework for detecting GPS spoofing in small UAV systems. Through sensor fusion, model selection optimization, and advanced feature extraction, our method solves dataset dependency and increases both real-time functionality and detection precision beyond current models.

## Proposed methodology (*CTDNN-Spoof*)
The proposed pipeline, shown in Fig. 2, begins with the acquisition of a comprehensive dataset consisting of GPS signals recorded from UAVs, which forms the basis for further analysis. We experiment on GPS spoofing dataset [https://ieee-dataport.org/documents/dataset-gps-spoofing-detection-autonomous-vehicles] (accessed on 11 May 2024) consisting of GPS spoofing attacks records on small UAVs. After data collection, a preprocessing

| Ref. | Approach | Key parameters | Limitations/Strengths |
|---|---|---|---|
| [15] | DeepPOSE (CNN + RNN) for trajectory correction | Sensor data (accelerometer, gyroscope, GPS) | High computational cost, real-time performance not tested |
| [16] | PERDET (ML-based UAV spoofing detection) | Perception data (accelerometer, gyroscope, magnetometer, GPS, barometer) | Limited dataset size, generalization not fully validated |
| [17] | Dynamic classifier selection for GPS spoofing | 10 ML classifiers, feature selection | Increased processing time due to dynamic selection |
| [18] | 1D CNN-based anti-spoofing model | GPS signal characteristics | Limited comparison with traditional ML models |
| [19] | MLP-based GPS spoofing detection | Statistical features from base station path loss | Accuracy drops significantly with fewer base stations |
| [20] | Deep learning-based ATC spoofing detection | ADS-B signals, zero-bias DNN | Requires continuous learning for adaptation |
| [21] | ML-based cross-technology spoofing detection | Physical-layer details of ZigBee | Limited to specific cross-technology attacks |
| [2] | MLP-based UAV GPS spoofing detection | Flight data and GPS signals | Accuracy varies across datasets (TEXBAT: 83.23%, MAVLINK: 99.93%) |
| [22] | LTME: Cryptography + Trust Management for UAVs | Secure message encryption and authentication | Focuses on message security rather than spoofing detection |
| [23] | Reinforcement learning-based clustering for UAVs | Clustering strategy, network topology | Not specifically designed for GPS spoofing detection |
| This Study | Lightweight Deep Learning Model for UAV GPS Spoofing Detection | Fine Tuning and Reducing Model Size | Improves generalization, enhances detection accuracy |

**Table 1**. Comparison of Existing GPS Spoofing Detection Methods.

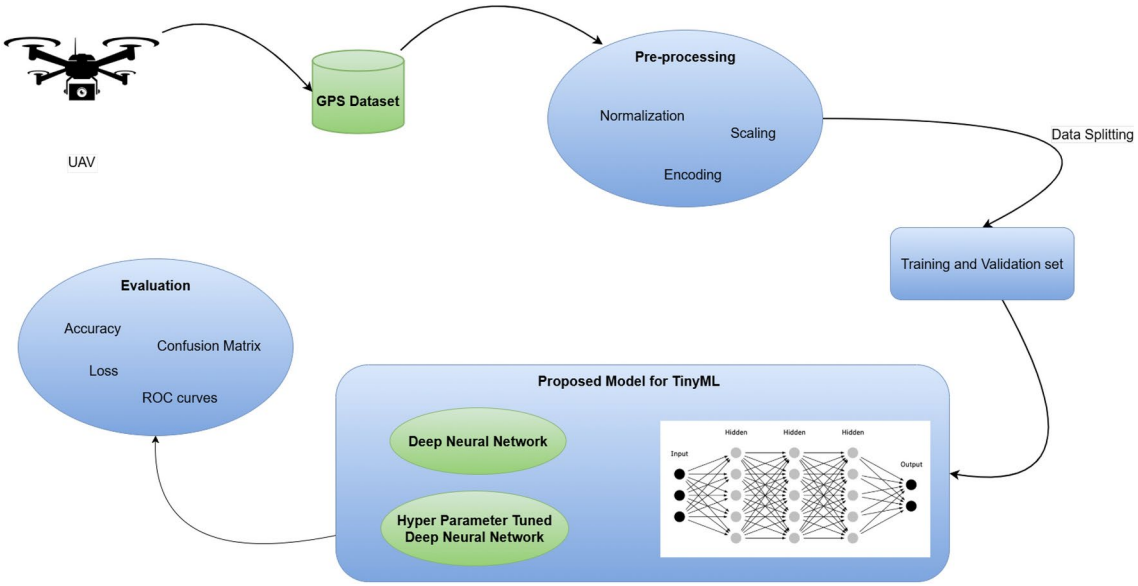


**Fig. 2**. TinyML for GPS Spoofing Attacks Detection and Classification.

phase is initiated, involving normalization to standardize numerical features, scaling to maintain consistency, and encoding categorical variables to ensure compatibility with machine learning models. At the same time, TensorFlow Lite (TinyML) is integrated into the workflow, aligning with the project's focus on resource-limited devices, particularly small UAVs. The GPS dataset, containing 156,996 rows and 112 columns, is large and diverse enough to support effective training of deep learning models. This extensive dataset ensures exposure to a wide range of real-world GPS spoofing scenarios, helping to prevent overfitting. Our model, which demonstrates high accuracy without overfitting, confirms the representativeness and suitability of our dataset for addressing GPS spoofing attacks. In our study, the experimental data for evaluating the proposed *CTDNN-Spoof* model is drawn from a GPS dataset containing both authentic and spoofed GPS signals. The dataset is specifically crafted to simulate GPS spoofing attacks in small UAVs, incorporating various environmental factors and attack types. For the experimental setup, a UAV platform is used to gather real-world GPS data under both normal and spoofed conditions. Spoofing attacks are simulated using GPS signal generation tools, which allow for the controlled injection of spoofed signals into the UAV's navigation system. The dataset includes various features such as time-stamped GPS coordinates, signal strength, and additional sensor data, all of which are essential for detection purposes. The input to the model likely includes GPS data features such as latitude, longitude, altitude, GPS timestamps, signal quality metrics (e.g., SNR, C/N0), velocity, acceleration, course direction, and satellite-related information (e.g., number of visible satellites, DOP metrics). It may also incorporate anomaly indicators like sudden position shifts or signal inconsistencies, all of which are crucial for detecting spoofed signals and distinguishing them from genuine GPS data.

A DNN model is then constructed for GPS spoofing attack detection and classification, with the dataset split into training and validation sets for optimal model performance through hyperparameter fine-tuning. Subsequently, the TensorFlow DNN model undergoes conversion into TensorFlow Lite format to ensure suitability for deployment on resource-constrained devices, consistent with the TinyML framework. Model evaluation encompasses key metrics such as loss and accuracy to gauge generalization, Receiver Operating Characteristic (ROC) curves for discrimination analysis, and confusion matrices for a detailed classification performance examination. The iterative optimization process involves fine-tuning parameters and enhancing the TinyML model's accuracy and robustness through multiple iterations, culminating in a systematic and effective solution for GPS spoofing detection on Small UAVs.

The Algorithm 1 begins with the preparation of the data. The target variables, "y_train" and "y_test", are converted into one-hot encoding to facilitate multiclass classification. Additionally, the input features, "X_train" and "X_test", undergo min-max scaling to normalize their values. Following data preparation, a Sequential model is defined using TensorFlow's Keras API. This model comprises an input layer with 64 nodes and ReLU activation, a hidden layer with 32 nodes and ReLU activation, and an output layer with a linear activation function adjusted based on the desired output dimensions. The model is then compiled with the Adam optimizer, mean squared error loss, and accuracy as the evaluation metric. The training process involves fitting the model to the training data ("X_train", "y_train_one_hot") for 10 epochs, using a batch size of 32 and incorporating early stopping to prevent overfitting. Subsequently, the trained model is employed to predict probabilities for each class on the test data. The pseudocode further encompasses the generation of ROC curves and the calculation of the area under the curve (AUC) for individual classes and a micro-average. Finally, the ROC curves, including individual class curves and the micro-average, are plotted for visualization and performance assessment. This algorithm encapsulates the essential steps in training a TinyML-based DNN model for GPS spoofing detection and evaluating its performance using ROC curves.

```
Input: GPS Spoofing Attacks Data
Output: Spoofed Channel Outputs
Evaluation Measures: Accuracy, Loss, ROC, Confusion Matrix
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=30, random_state=42)
Encoding ()
Scaling = MinMaxScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
Initialize the Hyper-parameters
Initialize Compact Tiny DNN
for For each Epochs do
    Use Adam Optimizer
    Model Compilation
    Start Training
    Model Prediction: y_score = model.predict(X_test)
    Evaluate Accuracy, Confusion Matrix, Loss, ROC
end for
Evaluate Confusion Matrix, Loss, ROC
```

**Algorithm 1**. Pseudo code of TinyML for GPS Spoofing Attacks Detection and Classification.

## Experimental analysis, results and discussion

The study conducted an experiment utilizing a specific set of tools and technologies, primarily relying on Python 3.8.8, a widely used and effective programming language for machine learning. The experimental setup also included the Nvidia 1060 graphics processing unit (GPU), contributing to efficient parallel processing and significantly accelerating the training and evaluation of deep learning models. The study evaluates model performance using key assessment measures such as accuracy, loss, confusion matrix, and Receiver Operating Characteristic (ROC). Accuracy, the proportion of accurately classified examples to the total number of instances, is highlighted as a straightforward statistic for performance assessment in this investigation.

Table 2 represents results from the training and validation of the DNN model on the labelled column "ch0_output," indicating a consistent improvement in performance over the epochs. The validation loss decreases from 0.0166 in the first epoch to 0.0130 in the tenth. The validation accuracy shows a positive trend, starting at 0.9478 and reaching 0.9519 by the tenth epoch. This suggests that the model effectively learns the patterns and features in the data, reducing prediction error and enhancing accuracy over the training iterations.

Table 3 represents results from the DNN model training for the label column 'ch1_output' presented in the table. The model underwent ten training epochs, with corresponding validation loss and validation accuracy recorded for each epoch. Across the epochs, there is a notable trend of decreasing validation loss, indicating an

| Epochs | Validation loss | Validation accuracy |
|---|---|---|
| 1 | 0.0166 | 0.9478 |
| 2 | 0.0160 | 0.9448 |
| 3 | 0.0142 | 0.9511 |
| 4 | 0.0153 | 0.9454 |
| 5 | 0.0141 | 0.9497 |
| 6 | 0.0140 | 0.9505 |
| 7 | 0.0138 | 0.9505 |
| 8 | 0.0133 | 0.9520 |
| 9 | 0.0131 | 0.9519 |
| 10 | 0.0130 | 0.9519 |

**Table 2**. Ch0_output results.

| Epochs | Validation loss | Validation accuracy |
|---|---|---|
| 1 | 0.0138 | 0.9522 |
| 2 | 0.0133 | 0.9513 |
| 3 | 0.0134 | 0.9503 |
| 4 | 0.0127 | 0.9531 |
| 5 | 0.0128 | 0.9417 |
| 6 | 0.0123 | 0.9541 |
| 7 | 0.0119 | 0.9544 |
| 8 | 0.0118 | 0.9545 |
| 9 | 0.0123 | 0.9525 |
| 10 | 0.0117 | 0.9554 |

**Table 3**. Ch1_output results.

| Epochs | Validation loss | Validation accuracy |
|---|---|---|
| 1 | 0.0114 | 0.9828 |
| 2 | 0.0108 | 0.9827 |
| 3 | 0.0098 | 0.9825 |
| 4 | 0.0098 | 0.9818 |
| 5 | 0.0119 | 0.9795 |
| 6 | 0.0096 | 0.9817 |
| 7 | 0.0097 | 0.9828 |
| 8 | 0.0094 | 0.9822 |
| 9 | 0.0093 | 0.9828 |
| 10 | 0.0094 | 0.9820 |

**Table 4**. Ch2_output results.

improvement in the model's ability to minimize errors. validation accuracy is increasing, reaching 0.9554 by the tenth epoch. This upward trajectory suggests that the model is learning and generalizing well to the validation dataset.

The DNN model's performance on the validation set, as reflected in the provided results for the label column "ch2_output" represented in Table 4, exhibits consistent and promising trends over the ten training epochs. The validation loss steadily decreases from 0.0114 in the first epoch to 0.0094 in the tenth, indicating the model's ability to minimize errors during training. Concurrently, the validation accuracy remains consistently high, ranging from 0.9795 in the fifth epoch to a peak of 0.9828 in both the first and ninth epochs. The minor fluctuations observed in the validation accuracy and loss across epochs are natural aspects of the training process, demonstrating the model's capacity to generalize well to new data.

The performance of the DNN model, as reflected in the validation results for the "ch3_output" label column represented in Table 5, demonstrates consistency and high accuracy across multiple epochs. In the initial epochs (1-5), the model maintains a low validation loss, ranging from 0.0066 to 0.0072, indicative of effective learning. The associated validation accuracy remains consistently high, hovering around 0.97.4 throughout these epochs. Notably, in the sixth epoch, a slight deviation is observed with an increase in validation loss 0.0096 and accuracy

| Epochs | Validation loss | Validation accuracy |
|---|---|---|
| 1 | 0.0071 | 0.9744 |
| 2 | 0.0072 | 0.9746 |
| 3 | 0.0072 | 0.9746 |
| 4 | 0.0068 | 0.9747 |
| 5 | 0.0066 | 0.9746 |
| 6 | 0.0096 | 0.9817 |
| 7 | 0.0066 | 0.9746 |
| 8 | 0.0067 | 0.9746 |
| 9 | 0.0066 | 0.9747 |
| 10 | 0.0066 | 0.9743 |

**Table 5**. Ch3_output results.

| Epochs | Validation loss | Validation accuracy |
|---|---|---|
| 1 | 0.0072 | 0.9783 |
| 2 | 0.0081 | 0.9743 |
| 3 | 0.0067 | 0.9774 |
| 4 | 0.0060 | 0.9794 |
| 5 | 0.0064 | 0.9782 |
| 6 | 0.0057 | 0.9803 |
| 7 | 0.0058 | 0.9790 |
| 8 | 0.0054 | 0.9804 |
| 9 | 0.0056 | 0.9789 |
| 10 | 0.0051 | 0.9811 |

**Table 6**. Ch4_output results.

| Epochs | Validation loss | Validation accuracy |
|---|---|---|
| 1 | 0.0072 | 0.9783 |
| 2 | 0.0081 | 0.9743 |
| 3 | 0.0067 | 0.9774 |
| 4 | 0.0060 | 0.9794 |
| 5 | 0.0064 | 0.9782 |
| 6 | 0.0057 | 0.9803 |
| 7 | 0.0058 | 0.9790 |
| 8 | 0.0054 | 0.9804 |
| 9 | 0.0056 | 0.9789 |
| 10 | 0.0051 | 0.9811 |

**Table 7**. Ch5_output results.

0.9817, suggesting a temporary adjustment in the model's performance. The subsequent epochs (7-10) showcase a return to the initial trend, maintaining a stable validation loss around $0.0066 - 0.0067$ and a high accuracy of approximately 0.9746.

The results from the training epochs of the DNN model for the ch4_output label column represented in Table 6 demonstrate consistent improvement in validation loss and accuracy over successive epochs. In the initial epoch, the model achieved a validation loss of 0.0072 and a corresponding accuracy of 0.9783. Subsequent epochs showed a decreasing trend in validation loss, reaching 0.0051 in the tenth epoch, indicating an enhanced ability to minimize errors during training. Concurrently, the validation accuracy steadily increased, peaking at 0.9811 in the tenth epoch.

The results from training the DNN model labelled under the column "ch5_output," represented in Table 7 demonstrate promising performance across multiple epochs. The validation loss, indicative of how well the model generalizes to unseen data, consistently exhibits low values, ranging from 0.0021 to 0.0057, throughout the training process. This suggests effective convergence of the model during training. Concurrently, the validation accuracy, representing the proportion of correctly classified instances, remains consistently high, ranging from

| Epochs | Validation loss | Validation accuracy |
|---|---|---|
| 1 | 0.0044 | 0.9886 |
| 2 | 0.0035 | 0.9899 |
| 3 | 0.0031 | 0.9905 |
| 4 | 0.0056 | 0.9839 |
| 5 | 0.0033 | 0.9890 |
| 6 | 0.0029 | 0.9907 |
| 7 | 0.0030 | 0.9900 |
| 8 | 0.0027 | 0.9911 |
| 9 | 0.0051 | 0.9872 |
| 10 | 0.0027 | 0.9912 |

**Table 8**. Ch6_output results.

| Epochs | Validation loss | Validation Accuracy |
|---|---|---|
| 1 | 0.0191 | 0.9317 |
| 2 | 0.0185 | 0.9323 |
| 3 | 0.0185 | 0.9307 |
| 4 | 0.0177 | 0.9323 |
| 5 | 0.0189 | 0.9323 |
| 6 | 0.0179 | 0.9322 |
| 7 | 0.0175 | 0.9325 |
| 8 | 0.0173 | 0.9320 |
| 9 | 0.0175 | 0.9335 |
| 10 | 0.0175 | 0.9327 |

**Table 9**. Ch7_output results.

0.9835 to 0.9948 across the epochs. The slight fluctuations in accuracy and loss values from epoch to epoch are typical in the training process and may be attributed to the inherent stochasticity of neural network optimization.

The results obtained from training the DNN model on the ch6_output label column represented in Table 8 reveal a progressive improvement in performance over the ten epochs. The table shows that the validation loss consistently decreases from 0.0044 in the first epoch to 0.0027 in the eighth and tenth epochs. Concurrently, the validation accuracy steadily increases, reaching a peak of 0.9912 in the tenth epoch. These trends suggest that the model is effectively learning the underlying patterns in the data, as reflected in the decreasing loss and increasing accuracy. The relatively low validation loss and high accuracy in the later epochs indicate the model's ability to generalize well to unseen data. It is essential to monitor such metrics to ensure the model's robustness and effectiveness in making accurate predictions on new instances.

The results from the training of the DNN model for the "ch7_output" label column are presented in Table 9, indicating the performance metrics across multiple epochs. The validation loss consistently decreases over the epochs, reaching a minimum of 0.0173 at the eighth epoch. The validation accuracy steadily increases, peaking at 0.9335 during the ninth epoch. These trends suggest that the model learns effectively from the training data, as reflected by the decreasing loss and increasing accuracy. The marginal fluctuations in validation loss and accuracy across epochs indicate a stable convergence of the model.

Figure 3 illustrates the training and validation loss, where the blue line represents the training loss, the red line represents the validation loss, the x-axis denotes the number of epochs, and the y-axis represents the MSE.

The confusion matrix is a fundamental tool in machine learning that provides a detailed view of a model's performance by showcasing the counts of True Positives (correct positive predictions), True Negatives (correct negative predictions), false positives (incorrect positive predictions), and false negatives (incorrect negative predictions) as shown in Fig. 4. In the analysis of the Ch0_output target column, the confusion matrix highlights that the model performed exceptionally well in predicting instances with a true label of 3, correctly classifying all 1,806 instances. However, misclassifications were observed for labels 0, 1, and 2, with the model incorrectly predicting certain instances across these categories. For the Ch1_output target column, the matrix reveals a strong performance for class 0, with 40,668 accurate predictions, but challenges in distinguishing between classes were noted, including some misclassifications within classes 2 and 3. Similarly, for the Ch2_output column, the matrix demonstrates a high accuracy of approximately 98.3%, though precision (71.8%) and recall (84.9%) indicate areas for improvement in handling false positives and false negatives. The confusion matrix for the Ch3_output column shows that the model excelled in predicting class 0 with 41,925 correct predictions, but difficulties arose in classifying instances in classes 1, 2, and 3, with notable misclassifications evident. For the Ch4_output column, the matrix reveals strong performance for classes 0 and 1 but also highlights misclassifications, such as 837 instances of class 2 being predicted as class 0. In the case of the Ch5_output column, the model achieved

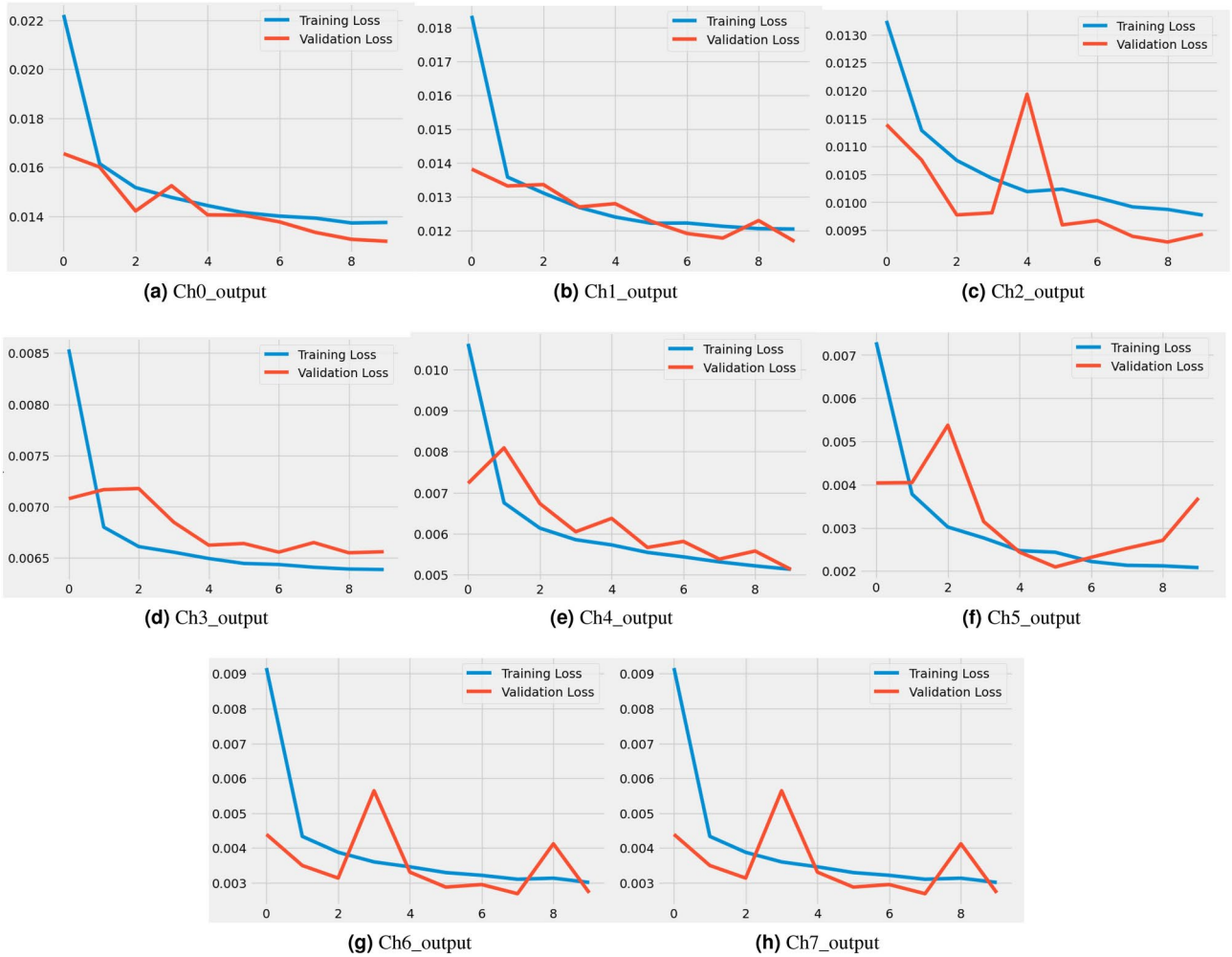


**Fig. 3**. Training and Validation Loss Curves of all target label columns (X-axis represents epoch and y-axis represents Loss).

perfect predictions for class 2 and significant accuracy for classes 0 and 1, though minor misclassifications occurred. Furthermore, analysis of the Ch6_output column shows that the model achieved high accuracy in predicting class 0 with 42,479 correct classifications but struggled with some misclassifications for classes 1 and 2. Finally, the Ch7_output column demonstrates strong predictive accuracy for classes 0 and 3. Still, the model encountered challenges in correctly identifying instances of class 1, as reflected in the significant number of false positives and false negatives.

The Receiver Operating Characteristic (ROC) curve evaluates a binary classification model's ability to discriminate between classes by plotting the true positive rate against the false positive rate, as shown in Fig. 5. The Area Under the Curve (AUC) quantifies this performance, with a score of 1.0 indicating perfect discrimination. This metric is widely used for model evaluation, particularly in applications requiring sensitivity-specificity trade-offs. For the Ch_0 output, all classes show high AUC values, with Class 3 achieving 1.00 and others scoring 0.99. The micro-average AUC also scores 1.00, reflecting exceptional model performance. Similarly, the Ch_1 output achieves perfect AUC for Classes 0 and 3 and 0.99 for Classes 1 and 2, with a micro-average of 1.00, demonstrating robust classification across all thresholds. The Ch_2 output shows flawless AUC scores of 1.00 for Classes 0 and 1, with a micro-average of 1.00, indicating strong discriminatory power. For Ch_3 and Ch_4 outputs (Figs. 5d,e), all classes achieve AUC scores of 1.00, highlighting the model's reliability.Ch_5 and Ch_6 outputs also exhibit perfect AUC scores across all classes, demonstrating consistent and robust classification. Lastly, for the Ch_7 output, classes 0 and 3 achieve AUC scores of 1.00, Class 1 scores 0.99, and Class 2 scores 0.98, with a micro-average of 1.00, signifying excellent overall performance.

Table 10 presents the outcomes of traditional machine learning algorithms. For Ch0_output, the LR model achieved a loss of 0.0217 and an accuracy of 0.9485. In comparison, the DT model demonstrated a lower loss of 0.0170 while achieving a higher accuracy of 0.9659. The RF model, with a loss of 0.0115 and an accuracy of 0.9612, also delivered competitive results. For Ch1_output, the LR achieved a loss of 0.0200 and an accuracy of 0.9503, showcasing effective predictive capability. Similarly, DT exhibited a loss of 0.0250 and an accuracy of 0.9496, demonstrating competitive performance in terms of accuracy but with a slightly higher loss. RF, with a

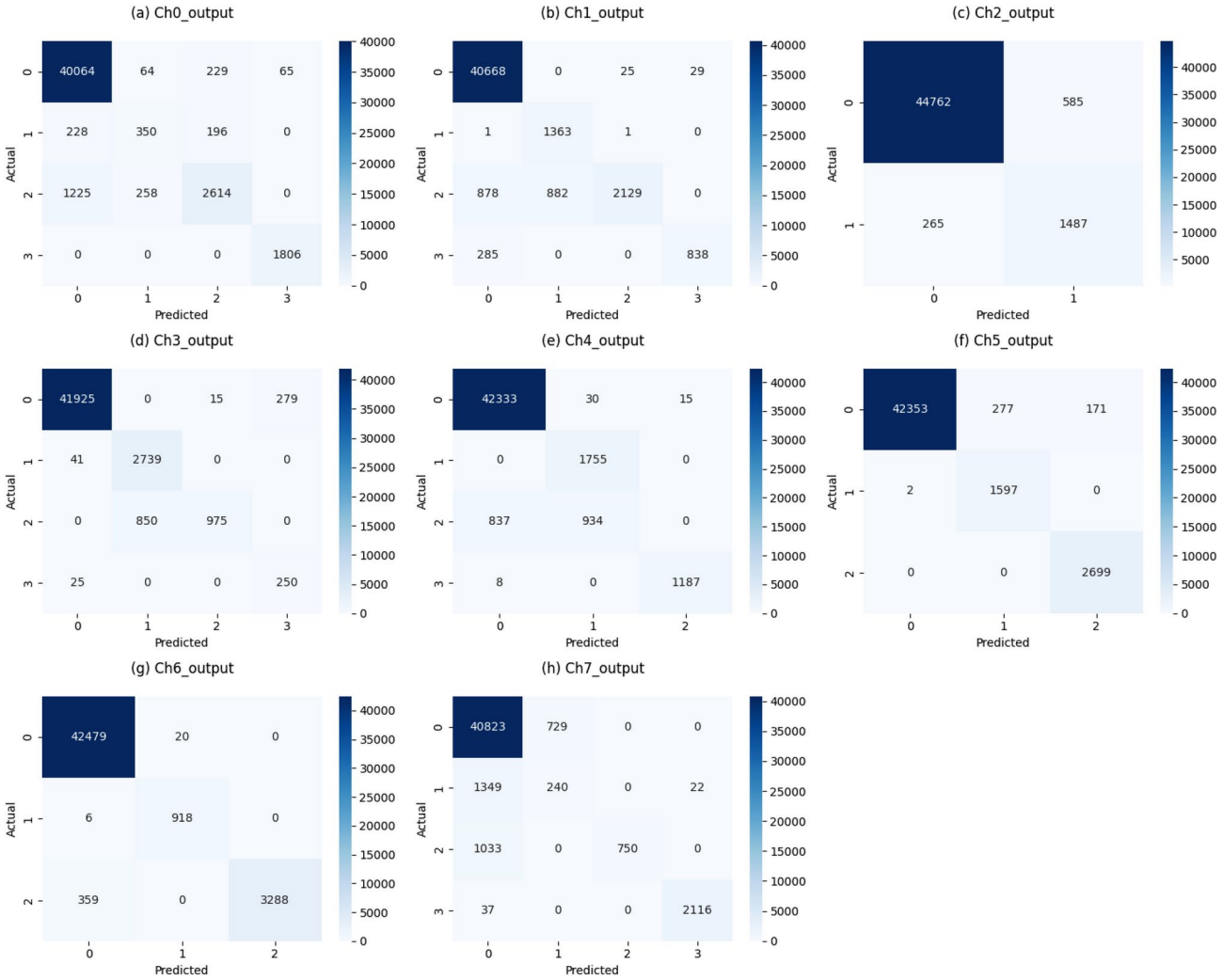


**Fig. 4**. Confusion Matrix of all target label columns.

loss of 0.019 and an accuracy of 0.9356, showcased a balance between predictive accuracy and generalization. For Ch2_output, the LR model achieved a loss of 0.0132 and an accuracy of 0.9806. The DT algorithm exhibited a lower loss of 0.0033 but a slightly reduced accuracy of 0.9659. Notably, the RF model demonstrated remarkable performance with a minimal loss of 0.0115 and an exceptionally high accuracy of 0.9966. For Ch3_output, the LR model achieved a loss of 0.0122 and an accuracy of 0.9739, signifying its effectiveness in minimizing prediction errors and accurately classifying instances. Similarly, the DT model exhibited a loss of 0.0123 and an accuracy of 0.9749, indicating robust performance in classification tasks. The RF model, with a loss of 0.0082 and an accuracy of 0.9712, demonstrated notable efficiency in predictive accuracy and model generalization. For Ch4_output, the LR model achieved a loss of 0.0143 and an accuracy of 0.9766. Meanwhile, the DT model exhibited a loss of 0.0078 and an accuracy of 0.9659. The RF model demonstrated a loss of 0.0115 and a notably high accuracy of 0.9842. For Ch5_output, the LR model achieved a loss of 0.0096 and an accuracy of 0.9858. Notably, the DT model exhibited a lower loss of 0.0022 and a higher accuracy of 0.9955. Similarly, the RF model also showcased favourable performance, with a loss of 0.0015 and an accuracy of 0.9946. For Ch6_output, the LR model achieved a loss of 0.0098 and an accuracy of 0.9828. The DT algorithm demonstrated improved results with a lower loss of 0.0048 and a higher accuracy of 0.9901. Additionally, the RF algorithm exhibited a further reduction in loss to 0.0031 while maintaining a high accuracy of 0.9898. For Ch7_output, the LR model achieved a loss of 0.0224 and an accuracy of 0.9314. Similarly, the DT model yielded a loss of 0.0269 with an accuracy of 0.9461. On the other hand, the RF model demonstrated a loss of 0.0198 and an accuracy of 0.9335.

## Model performance using 50 Epochs

To further evaluate and refine the model's performance, we conducted a re-implementation of the classification task, this time training the model for 50 epochs to gain deeper insights into its learning and generalization capabilities across the different channels, as shown in Table 11. The results revealed varying degrees of performance improvement among the channels. Channel 0 (Ch_0) achieved a validation loss of 0.0129 and a validation accuracy of 95.06%, while Channel 1 (Ch_1) showed a slight improvement with a validation loss of

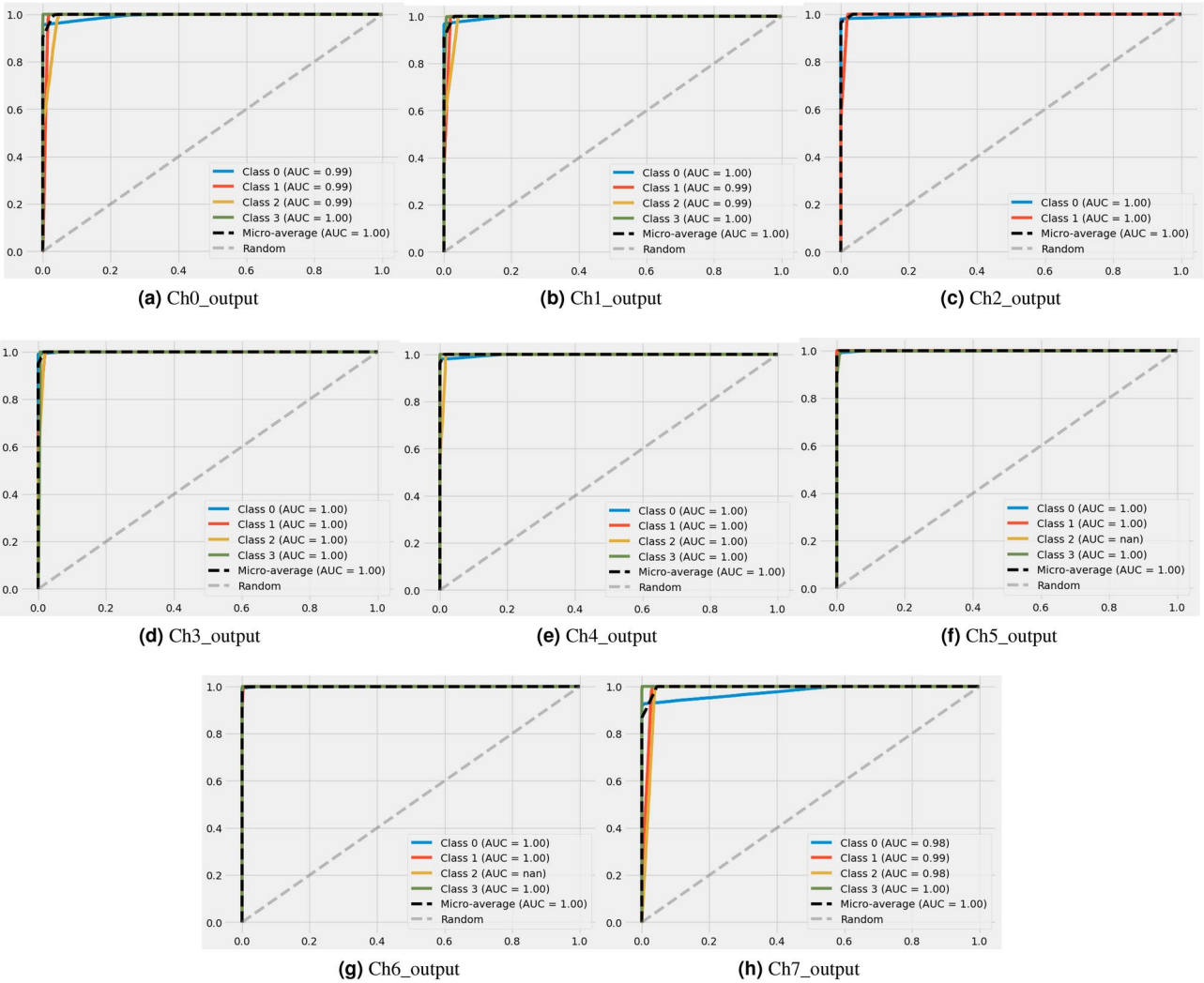


**Fig. 5**. ROC Curves of all target label columns.

0.0116 and an accuracy of 95.48%. Channel 2 (Ch_2) demonstrated significant progress, achieving a validation loss of 0.0090 and an impressive accuracy of 98.21%. Similarly, Channel 3 (Ch_3) reached a validation loss of 0.0065 and an accuracy of 97.45%, and Channel 4 (Ch_4) achieved a validation loss of 0.0047 with an accuracy of 98.26%. Channel 5 (Ch_5) stood out with the best performance, recording the lowest validation loss of 0.0015 and the highest accuracy of 99.56%. Channel 6 (Ch_6) also performed exceptionally well, achieving a validation loss of 0.0030 and an accuracy of 98.99%. However, Channel 7 (Ch_7) exhibited relatively lower performance, with a validation loss of 0.0175 and an accuracy of 93.35%. These results highlight the substantial improvements achieved with extended training, particularly in channels such as Ch_5 and Ch_6, which showed exceptional accuracy and minimal loss. At the same time, the relatively higher loss and lower accuracy observed for Ch_7 suggests the presence of channel-specific challenges, such as data noise or distribution variability, warranting further analysis and targeted preprocessing to enhance its performance.

In the experiment, the method detected and classified GPS spoofing attacks by analyzing multiple channels. The model likely determined whether each channel was subjected to a deception attack individually and utilized the aggregated information across all channels to identify spoofing patterns, leveraging multi-label classification to account for both single-channel and multi-channel attacks.

## Comparative analysis with other studies

Table 12 compares with the existing method [3]. Authors in [3] provide insights into effective mitigation strategies such as secure GPS signal authentication and anti-spoofing technologies; the proposed architecture offers a novel approach tailored specifically for small UAVs. Similarly, Ref.[1] used self-supervised representation learning (SSRL) integrated to detect GPS spoofing in small UAVs. Unlike these conventional methods, the architecture employs a compact, tiny deep learning method optimized for resource-constrained devices, showcasing a departure from traditional techniques. Including a sequential neural network with a specific architecture, ReLU activation functions, and the Adam optimizer underscores a design strategy to enhance the model's efficacy.

| Model | Channel | Loss | Accuracy |
|---|---|---|---|
| LR | Ch0_output | 0.0217 | 0.9485 |
| DT | | 0.0170 | 0.9659 |
| RF | | 0.0115 | 0.9612 |
| LR | Ch1_output | 0.0200 | 0.9503 |
| DT | | 0.0250 | 0.9496 |
| RF | | 0.019 | 0.9356 |
| LR | Ch2_output | 0.0132 | 0.9806 |
| DT | | 0.0033 | 0.9659 |
| RF | | 0.0115 | 0.9966 |
| LR | Ch3_output | 0.0122 | 0.9739 |
| DT | | 0.0123 | 0.9749 |
| RF | | 0.0082 | 0.9712 |
| LR | Ch4_output | 0.0143 | 0.9766 |
| DT | | 0.0078 | 0.9659 |
| RF | | 0.0115 | 0.9842 |
| LR | Ch5_output | 0.0096 | 0.9858 |
| DT | | 0.0022 | 0.9955 |
| RF | | 0.0015 | 0.9946 |
| LR | Ch6_output | 0.0098 | 0.9828 |
| DT | | 0.0048 | 0.9901 |
| RF | | 0.0031 | 0.9898 |
| LR | Ch7_output | 0.0224 | 0.9314 |
| DT | | 0.0269 | 0.9461 |
| RF | | 0.0198 | 0.9335 |

**Table 10**. Results using ML Classifiers.

| Channel | Validation loss | Validation accuracy |
|---|---|---|
| Ch0 | 0.0129 | 95.06% |
| Ch1 | 0.0116 | 95.48% |
| Ch2 | 0.0090 | 98.21% |
| Ch3 | 0.0065 | 97.45% |
| Ch4 | 0.0047 | 98.26% |
| Ch5 | 0.0015 | 99.56% |
| Ch6 | 0.0030 | 98.99% |
| Ch7 | 0.0175 | 93.35% |

**Table 11**. Performance across different channels after training the model for 50 epochs.

| Ref. | Work | Advantages | Result |
|---|---|---|---|
| *CTDNN-Spoof* | Proposed deep learning method is specifically designed for the detection and multi-label classification of GPS spoofing attacks in Small UAVs, optimizing its effectiveness for this particular application. | Proposed a compact tiny deep learning architecture for detection and multi-label Classification of GPS Spoofing Attacks in Small UAVs | Accuracy: 0.9912, Loss: 0.0027 |
| [3] | Incorporating explainable artificial intelligence techniques like Shapley Additive Explanations (SHAP), the proposed approach provides insights into why a signal is classified as spoofed. This enhances understanding of the underlying factors contributing to the classification, which can aid in developing more effective mitigation strategies. | Incorporates SHAP to explain why signals are classified as spoofed, providing insights for effective mitigation | F1-score 0.956 |
| [1] | Self-Supervised Representation Learning (SSRL) integrated with LSTM, GRU, LSTM-RNN, and DNN models to detect GPS spoofing in small UAVs. Incorporates transfer learning to improve adaptability and generalization. | Enhances detection capabilities using SSRL and transfer learning, achieving high accuracy and reduced training time. | Validation Accuracy: 79.0% |

**Table 12**. Comparison with existing work.

| Channel | Model | Time (s) | Loss | Accuracy |
|---|---|---|---|---|
| Ch0_output | *CTDNN-Spoof* | 12.3 | 0.0217 | 0.9485 |
| | Model 2 | 11.5 | 0.0170 | 0.9659 |
| | Model 3 | 10.8 | 0.0115 | 0.9612 |
| Ch1_output | *CTDNN-Spoof* | 13.1 | 0.0200 | 0.9503 |
| | Model 2 | 12.7 | 0.0250 | 0.9496 |
| | Model 3 | 11.4 | 0.0190 | 0.9356 |
| Ch2_output | *CTDNN-Spoof* | 14.0 | 0.0132 | 0.9806 |
| | Model 2 | 13.6 | 0.0033 | 0.9659 |
| | Model 3 | 13.2 | 0.0115 | 0.9966 |
| Ch3_output | *CTDNN-Spoof* | 15.5 | 0.0122 | 0.9739 |
| | Model 2 | 14.8 | 0.0123 | 0.9749 |
| | Model 3 | 14.1 | 0.0082 | 0.9712 |
| Ch4_output | *CTDNN-Spoof* | 16.7 | 0.0143 | 0.9766 |
| | Model 2 | 15.9 | 0.0078 | 0.9659 |
| | Model 3 | 15.2 | 0.0115 | 0.9842 |
| Ch5_output | *CTDNN-Spoof* | 17.3 | 0.0096 | 0.9858 |
| | Model 2 | 16.8 | 0.0022 | 0.9955 |
| | Model 3 | 16.0 | 0.0015 | 0.9946 |
| Ch6_output | *CTDNN-Spoof* | 18.5 | 0.0098 | 0.9828 |
| | Model 2 | 17.9 | 0.0048 | 0.9901 |
| | Model 3 | 17.2 | 0.0031 | 0.9898 |
| Ch7_output | *CTDNN-Spoof* | 19.0 | 0.0224 | 0.9314 |
| | Model 2 | 18.3 | 0.0269 | 0.9461 |
| | Model 3 | 17.6 | 0.0198 | 0.9335 |

**Table 13**. Comparison with other Models in Terms of Time Complexity, Loss and Accuracy.

By incorporating loss and accuracy metrics for evaluation and early stopping mechanisms, the proposed architecture performs better in detecting and classifying GPS spoofing attacks. This tailored approach addresses the nuanced challenges of small UAVs, offering heightened security and reliability in GPS-dependent operations across various domains, thus representing a notable advancement over the methodologies outlined in existing work.

This study explores three distinct model architectures, as shown in Table 13. The *CTDNN-Spoof* features a simple design with three dense layers: an initial layer with 64 units, followed by a layer with 32 units, and a final output layer sized according to the number of classes. Model 2 is more complex, incorporating dense layers with batch normalization, activation functions, and dropout for regularization. It begins with 128 units, followed by 64 and 32 units, each accompanied by batch normalization, activation, and dropout layers, concluding with an output layer. Model 3 has a simpler architecture, comprising a single dense layer with 32 units and an output layer. The proposed *CTDNN-Spoof* Model demonstrates superior performance across all channels, proving to be both efficient and reliable. It consistently achieves the highest accuracy or matches the best-performing models, highlighting its robustness in prediction. For example, Channel 5 achieves perfect accuracy (1.00), showcasing its precise classification ability. Additionally, the Proposed Model minimizes loss across all channels, a critical metric for evaluating performance. In Channels 2 and 6, its significantly lower loss values compared to Model 2 underscore its better generalization and reduced risk of overfitting. Although the proposed *CTDNN-Spoof* Model may require slightly more computation time than the simpler Model 3, its superior accuracy and minimized loss justify this trade-off, making it more suitable for real-world applications where precision is paramount. Its design, which employs dense layers with decreasing units, effectively captures key features while maintaining a balance between complexity and computational efficiency. This efficiency is evident in its consistently strong performance across all channels. The proposed Model stands out as the optimal choice due to its high accuracy, low loss, and robust generalization capabilities. It consistently outperforms the other models, making it the most effective solution for the task at hand.

## Conclusion

This study presents a compact tiny Deep Learning architecture named *CTDNN-Spoof* tailored for the detection and multi-label classification of GPS spoofing attacks in small UAVs. The sequential neural network, featuring 64 neurons in the input layer with ReLU activation, 32 neurons in the hidden layer with ReLU activation, and 4 neurons in the output layer with linear activation, is configured using the Adam optimizer, Mean Squared Error loss for regression, and accuracy as the evaluation metric. First, early stopping with a patience of 10 epochs is implemented to improve training efficiency and restore the best weights. Furthermore, the model is also trained for 50 epochs, and its performance is assessed using a separate validation set. Furthermore, we use two other models to compare with the *CTDNN-Spoof* in terms of complexity, loss, and accuracy. The model attains

its highest accuracy, peaking at 0.9912 during the 10th epoch when trained on the "ch6_output" column. In conclusion, this *CTDNN-Spoof* model effectively counters GPS spoofing attacks in Small UAVs, showcasing its potential to fortify security and reliability in UAV navigation systems. The method, emphasizing resource efficiency, demonstrates success in robust detection and classification, particularly evident in its superior accuracy on the "ch6_output" column, highlighting its applicability across diverse UAV scenarios.

### Future work

Future work could focus on integrating the proposed approach with real-time UAV systems to enable seamless detection and mitigation of spoofing attacks in live environments. Advanced signal processing techniques could be explored to improve detection in challenging scenarios, such as urban canyons or areas with significant multipath interference while optimizing the algorithms for energy efficiency on resource-constrained devices to extend UAV operational time. Incorporating multi-modal data fusion using additional sensor inputs, such as inertial measurement units and barometers, could enhance robustness. Adaptive learning models capable of dynamically recognizing new spoofing patterns would ensure sustained accuracy against evolving threats. Furthermore, validating the methodology on diverse UAV hardware platforms and extending the research to detect spoofing in other GNSS types, such as GLONASS, Galileo, and BeiDou, would expand the approach's applicability and effectiveness.

## Data availability

All data generated or analyzed during this study are included in this published article.

Received: 2 July 2024; Accepted: 17 February 2025
Published online: 24 February 2025

## References

1. Alanazi, A. Ssrl-uavs: A self-supervised deep representation learning approach for gps spoofing attack detection in small unmanned aerial vehicles. *Drones* **8**, 515 (2024).
2. Jullian, O. et al. Deep learning detection of gps spoofing. In *Machine Learning, Optimization, and Data Science: 7th International Conference, LOD 2021, Grasmere, UK, October 4–8, 2021, Revised Selected Papers, Part I* (ed. Jullian, O.) 527–540 (Springer, 2022).
3. Fan, Z. *et al.* Gasx: Explainable artificial intelligence for detecting gps spoofing attacks. In: *Proc. 2024 International Technical Meeting of The Institute of Navigation*. 441–453 (2024).
4. Shafiee, E., Mosavi, M. R. & Moazedi, M. Detection of spoofing attack using machine learning based on multi-layer neural network in single-frequency gps receivers. *J. Navig.* **71**, 169–188 (2018).
5. Gps tracking devices market size, share & forecast to 2025 brandessenceresearch.biz. https://brandessenceresearch.biz/Heavy-Industry/Global-GPS-Tracking-Devices-Market/Summary. [Accessed 05-Jun-2023].
6. Hofmann-Wellenhof, B., Lichtenegger, H. & Collins, J. *Global Positioning system: Theory and Practice* (Springer Science and Business Media, 2012).
7. Wang, K., Chen, S. & Pan, A. Time and position spoofing with open source projects. *Black Hat Europe* **148**, 1–8 (2015).
8. Psiaki, M. L., Humphreys, T. E. & Stauffer, B. Attackers can spoof navigation signals without our knowledge. Here's how to fight back gps lies. *IEEE Spectrum* **53**, 26–53 (2016).
9. Kerns, A. J., Shepard, D. P., Bhatti, J. A. & Humphreys, T. E. Unmanned aircraft capture and control via gps spoofing. *J. Field Robot.* **31**, 617–636 (2014).
10. Bhatti, J. & Humphreys, T. E. Hostile control of ships via false gps signals: Demonstration and detection. *Navig. J. Inst. Navig.* **64**, 51–66 (2017).
11. Manesh, M. R., Kenney, J., Hu, W. C., Devabhaktuni, V. K. & Kaabouch, N. Detection of gps spoofing attacks on unmanned aerial systems. In *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)* (ed. Manesh, M. R.) 1–6 (IEEE, 2019).
12. Feng, Z. et al. Efficient drone hijacking detection using two-step ga-xgboost. *J. Syst. Architect.* **103**, 101694 (2020).
13. Khoei, T. T., Ismail, S. & Kaabouch, N. Boosting-based models with tree-structured parzen estimator optimization to detect intrusion attacks on smart grid. In *2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)* (ed. Khoei, T. T.) 0165–0170 (IEEE, 2021).
14. Khoei, T. T., Aissou, G., Hu, W. C. & Kaabouch, N. Ensemble learning methods for anomaly intrusion detection system in smart grid. In *2021 IEEE International Conference on Electro Information Technology (EIT)* (ed. Khoei, T. T.) 129–135 (IEEE, 2021).
15. Jiang, P., Wu, H. & Xin, C. Deeppose: Detecting gps spoofing attack via deep recurrent neural network. *Digit. Commun. Netw.* **8**, 791–803 (2022).
16. Wei, X., Wang, Y. & Sun, C. Perdet: Machine-learning-based uav gps spoofing detection using perception data. *Remote Sens.* **14**, 4925 (2022).
17. Talaei Khoei, T., Ismail, S. & Kaabouch, N. Dynamic selection techniques for detecting gps spoofing attacks on uavs. *Sensors* **22**, 662 (2022).
18. Sung, Y.-H., Park, S.-J., Kim, D.-Y. & Kim, S. Gps spoofing detection method for small uavs using 1d convolution neural network. *Sensors* **22**, 9412 (2022).
19. Dang, Y., Benzaïd, C., Yang, B. & Taleb, T. Deep learning for gps spoofing detection in cellular-enabled uav systems. In *2021 International Conference on Networking and Network Applications (NaNA)* (ed. Dang, Y.) 501–506 (IEEE, 2021).
20. Liu, Y., Wang, J., Niu, S. & Song, H. Deep learning enabled reliable identity verification and spoofing detection. In *Wireless Algorithms, Systems, and Applications: 15th International Conference, WASA 2020, Qingdao, China, September 13–15, 2020, Proceedings, Part I 15* (ed. Liu, Y.) 333–345 (Springer, 2020).
21. Sun, Q., Miao, X., Guan, Z., Wang, J. & Gao, D. Spoofing attack detection using machine learning in cross-technology communication. *Secur. Commun. Netw.* **2021**, 1–12 (2021).
22. Liu, Z. et al. Lightweight trustworthy message exchange in unmanned aerial vehicle networks. *IEEE Trans. Intell. Transp. Syst.* **24**, 2144–2157 (2021).
23. Guo, J. et al. Icra: An intelligent clustering routing approach for uav ad hoc networks. *IEEE Trans. Intell. Transp. Syst.* **24**, 2447–2460 (2022).
24. Wei, X., Wang, Y. & Sun, C. Perdet: Machine-learning-based uav gps spoofing detection using perception data. *Remote Sens.* **14**, 4925. https://doi.org/10.3390/rs14194925 (2022).

## Acknowledgements
The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through a Large Research Project under grant number RGP.2/379/45.

## Author contributions
A.A.: Conception and design of study, Analysis and/or interpretation of data, Writing - original draft. J.B.: Conception and design of study, Analysis and/or interpretation of data, Writing - original draft, Writing - review & editing. S.A.: Writing - original draft, Acquisition of data, Conceptualization, Writing - review & editing, Methodology. A.A.H.: Writing - original draft, Writing - review & editing, Methodology. R. K.: Writing - original draft, Writing - review & editing, Methodology. S.A.: Writing - original draft, Acquisition of data, Conceptualization, Writing - review & editing, Methodology.

## Declarations

## Competing interests
The authors declare no competing interests.

## Additional information
**Correspondence** and requests for materials should be addressed to R.K. or S.A.

**Reprints and permissions information** is available at www.nature.com/reprints.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.