ISSN: 3007-3391

Journal of Computers and Intelligent Systems
Vol. 3, No. 3, 2025, pages 195 – 205
Journal Homepage:
https://journals.iub.edu.pk/index.php/JCIS/

Advanced Malicious Behavior Classification Using a Refined ANN-CNN Model

Humza Rana[1]
[1]Bahauddin Zakariya University Multan, Pakistan

ARTICLE INFO

Article History:

Received:	June	04, 2025
Revised:	August	05, 2025
Accepted:	August	10, 2025
Available Online:	August	25, 2025

Keywords:

Malware Classification
Deep Neural Network
Artificial Intelligence
Intrusion Detection

Classification Codes:

Funding:

This research received no specific grant from any funding agency in the public or not-for-profit sector.

ABSTRACT

The fastest-growing part of the network has led to an overwhelming increase in online data. Activities such as data transfer, online banking, and business transactions are now conducted over the internet, which, while providing convenience, also presents opportunities for malware developers to exploit vulnerabilities. Hackers use advanced techniques to break security measures, stealing personal data and demanding a ransom from victims. To overcome these increasing threats, there is an important need for more advanced AI-based methods to detect and prevent malware attacks.

In this paper, we propose an improved hybrid ANN-CNN sequential model designed to enhance malware classification performance. The category inequality is solved using the SMOTE method, which confirms that all categories are equally represented. Moreover, Principal Component Analysis (PCA) is employed for feature selection, enabling the model to focus on the most useful features and improving both training efficiency and model accuracy.

The model is evaluated on three multiclass datasets: WSN (Wireless Sensor Network), Microsoft Malware, and Virus Malware Digit. The presented architecture attained 98.1%, 99.6%, and 99.0% accuracy, respectively, demonstrating its effectiveness in handling complex, imbalanced, and diverse malware datasets.

Corresponding Author's Email: Humza.Rana99@gmail.com

Citation:

1. Introduction

Malicious software (malware) refers to any program or file designed to disrupt the normal operation of computer systems, cause damage, or gain unauthorized access to sensitive information. Malware is spreading rapidly, often to steal personal or confidential data and demand ransom payments from victims [1]. As cyberattacks grow in scale and sophistication, they have become a global concern, posing serious threats to businesses, organizations, and individuals. Cybercriminals often exploit system vulnerabilities to steal data, disrupt operations, or blackmail users.

Traditional malware detection methods, such as signature-based and dynamic analysis techniques, have notable limitations. They often fail to detect new or evolving types of malware, making them less effective against modern threats. As a result, there is an increasing need for advanced machine learning and deep learning techniques to accurately identify and classify malware [2].

For example, Langfang et al. (2022) [3] introduced a CNN-based model called MalShuffleNet, which classifies malware using the Malimg dataset. However, to improve classification performance, especially on datasets with uneven class distributions, it is important to integrate class imbalance handling techniques, such as SMOTE, to ensure more accurate predictions across all classes. Additionally, feature selection methods can help reduce dimensionality and improve model efficiency and training time.
Similarly, Imtiaz et al. (2021) [4] proposed DeepAMD, a deep ANN model for malware detection in Android systems, achieving an accuracy of 93.4%. While promising, there is still room for improvement, particularly when dealing with multiclass datasets.
Wei et al. (2023) [5] developed an intrusion detection model using 1D-CNN, combined with borderline SMOTE, Gaussian Mixture Model (GMM), and Quantum Particle Swarm Optimization (QPSO). Their method, named BSGM, achieved 99.95% accuracy on the KDD dataset with five classes. Despite the high accuracy, the issue of class imbalance still persisted, indicating the need for more robust balancing techniques.
In this paper, an improved hybrid ANN-CNN sequential model is proposed. Enhancements include:

- Increasing the number of layers for deeper learning,
- Applying class imbalance techniques (like SMOTE) to ensure fair learning across all classes,
- Using feature selection (such as PCA) to reduce model complexity and improve computational efficiency.

These improvements aim to enhance classification accuracy, especially on challenging multiclass malware datasets.

A. Problem

In deep learning, especially in detection works, category inequality is a popular problem in which some categories have significantly more samples than others. This inequality can lead to poor generalization, meaning the model might perform well on the majority class but poorly on the minority classes. As outcome, it can direct to wrong predictions, especially for the underrepresented classes, and can increase the complexity of model training because the model struggles to learn a balanced representation of all classes.

When dealing with multiclass datasets, this imbalance can cause the model to become biased, favoring the classes with more data. This makes it difficult for the architecture to accurately differentiate between less-represented categories.

To reduce these problems, feature selection becomes important. By taking the most useful characteristics, you can:

- Decrease the model’s complexity,
- Limited training time,
- Increase performance by terminating noisy or meaningless data.

Moreover, adapting the quantity of layers in a neural network (or the complexity of the model in general) is crucial. Too many layers might lead to overfitting, especially with inequality data, while too few may not capture enough patterns in the data. So, tuning the network architecture helps in improving classification accuracy and efficiency.

B. Contribution

In this paper, the SMOTE (Synthetic Minority Over-sampling Technique) technique is implemented to three different multiclass datasets to solve the category inequality problem. SMOTE works by generating synthetic samples for the minority categories, facilitating to maintenance of in equal number of instances across all categories. This improves the model's ability to generalize and reduces bias toward the majority classes.

To further increase model performance, Principal Component Analysis (PCA) is used as a feature selection and dimensionality reduction technique. By taking only the most important features, PCA helps reduce the complexity of the model, making the training process faster and more efficient while potentially improving accuracy.

Moreover, the architecture of the ANN-CNN hybrid model is optimized by adjusting the number of layers and tuning various hyperparameters (like learning rate, batch size, activation functions, etc.). This shaping is done to attain the best possible performance when classifying the data from these multiclass datasets.

2. Related Work

Rana et al. 2023 [6] proposed an ANN model to detect the malware. The proposed model deals with large datasets of APIs in a multiclass. The multiclass model achieves the best performance, with a 99.6% accuracy. The further class imbalance and model layers should be incorporated in it to achieve accurate performance and reduce the model computation. Rana et al. 2024 [7] proposed an ANN-CNN hybrid model to classify the malicious intrusion. The multiclass model achieves the best accuracy in the multiclass dataset. The class imbalance and feature selection should be implemented in it to achieve the best and efficient classification by the model.

Almousa et al. 2023 [8] proposed a character-awareness type language-based model to detect the semantic social attacks in URLs. Three models implemented in it include LSTM, CNN, and a character Bert-type model. The URL-based datasets used in this experiment contain five classes. After experimenting with three models, the character Bert achieves a 99.9 % accuracy performance. Iqbal et al. 2022 proposed an LSTM model for predicting malware. The PE imports dataset is used in this experiment. The dataset contains two malware classes. After an experiment, the proposed model achieved a 99.6% accuracy performance.

Altunay et al. 2021 [9] proposed a CNN model for detecting network intrusion. The CSE-CIC-IDS dataset was used in this experiment. The SMOTE technique dataset used in this experiment is used to balance the classes in the dataset. The dataset contains six intrusion types. After an experiment 98.8% accuracy performance was achieved. Gupta et al. 2022 [10] proposed an ANN model to classify the malware. The BIG dataset 2015 was used in this experiment. The dataset contains ten malware classes. Their proposed ANN model achieves 90.17% accuracy performance. The further CNN model needs to be implemented to enhance the accuracy performance in classifying the malware attacks.

Kinkead et al. 2021 [11] proposed a CNN model for detecting malware in the Android system. The Derbin benchmark dataset was used in this experiment. The LIME method was used in this experiment to extract the important features by measuring the activations. The dataset contains 5560 malicious apps from different types of malware. Their proposed model achieves 0.98% accuracy performance. Dwi et al. 2021[12] proposed ANN model for detect the ransomware in Bitcoin. The proposed model achieved a 97% accuracy performance. The model detection system is developed in Weka software using a backpropagation ANN model. In the future the variables should be added to enhance the performance in Weka software.

Fu et al. 2021 [13] proposed an LSTM model for detecting malware using transfer learning. The Android apps dataset used in this experiment contains the malicious and benign apps. The model achieves a 99.9% accuracy performance. The large dataset needs to train this model for detect the malware. The multiclass datasets need to be implemented in this model. The cloud-based malware detection needs to be considered. Sharma et al. 2021 [14] proposed ANN-GWO for detecting network intrusion. The MIT DARPA dataset was used in this experiment. The dataset contains the four intrusion types. After an experiment, 98.2% accuracy was obtained from the model. The additional multiclass dataset needs to be implemented in this model.

Table 1 Comparison of Different Approaches with Proposed Model

No	Model	Dataset	Accuracy
1	ANN-GWO	MIT DARPA	98.2%
2	CNN	Derbin Dataset	0.98%
3	ANN	Big dataset	0.9017%
4	Proposed Model	WSN, Microsoft Malware, Virus Digits	0.996%, 0.981%, 0.99%

Table 1 shows different types of deep learning methods for classifying malware attacks and network intrusions. The table states the proposed model, which is a hybrid that performs excellently in malware detection or intrusion detection in different types of classes in classification. In the proposed model, which is a hybrid model that combines ANN and CNN, the architectures combine to make a powerful classification. In data preprocessing, PCA is used to reduce the extra and high dimensionality in data to extract meaningful features from the dataset. The SMOTE also helps in handling the class imbalance issue, which makes our approach suitable for detecting malware and intrusion with the highest accuracy.

3. Used Approach

The proposed approach combines the ANN and CNN models into a hybrid model that improves and increases performance in classification tasks, especially dealing with multiple classes which are inequivalent labels. The proposed approach is discussed below.

A. Data Preprocessing and Class Imbalance Problem

Prepare the necessary setup and import essential libraries for deep learning, data preprocessing, evaluation, and visualization. Set environment settings to suppress unnecessary warnings and logs for cleaner output. Load and Preprocess the Dataset. Read the dataset using a CSV file loader. Extract the target column (Y = df['Class']) and drop it from the feature set. Normalize Features applies Min-Max Scaling to normalize the feature values between 0 and 1. Handle Class Imbalance Using SMOTE. Apply the Synthetic Minority Oversampling Technique (SMOTE) to balance the dataset classes. Use K-Neighbors value 3.

B. Dimensionality Reduction with PCA

To reduce the dimensionality of features, PCA is used in this approach. The extra dimensionality reduces the model performance and time performance in the model for performing on less data. Reduce dataset dimensions to 20 components using Principal Component Analysis (PCA).

C. Labels Encoding and Shaping Data

Encode Target Labels. One-hot encode the resampled labels. Split Data into Training, Validation, and Testing Sets**.** Split the balanced data into training, validation, and testing subsets. Reshape Data for CNN Input**.** Reshape the feature set (X) to the format required by 1D CNN layers (e.g., adding a third dimension).

D. Hybrid architecture design

Define the ANN-CNN Model. Use a Sequential model to build the hybrid architecture.

1. **CNN Layers**:

- ✓ Add Conv1D layers with ReLU activation to extract spatial patterns.
- ✓ Use MaxPoolingID layers for dimensionality reduction.

2. **Flatten Layer**:

✓ Flatten the extracted features to feed into the ANN layers.

3. **ANN Layers**:

✓ Add fully connected Dense layers with ReLU activation.
✓ Add Dropout layers to prevent overfitting.

4. **Output Layer**:

✓ Add a Dense layer with SoftMax activation for multiclass classification. This layer determines the number of classes that will be used in classification.

E. Compile the Model

Compile the model with: RMSProp optimizer. Categorical cross-entropy loss. Binary accuracy as a performance metric. Generate Model Diagram: Visualize and save the architecture diagram of the model. Train the Model: Train the model with Training data, validation data. 1000 epochs and a batch size of 3500. Track the training time.

F. Evaluate the Model

Test the model's performance on the test dataset. Record and display evaluation metrics such as accuracy, confusion matrix, F1 score, precision, recall, and classification report.

G. Hybrid Architecture Diagram

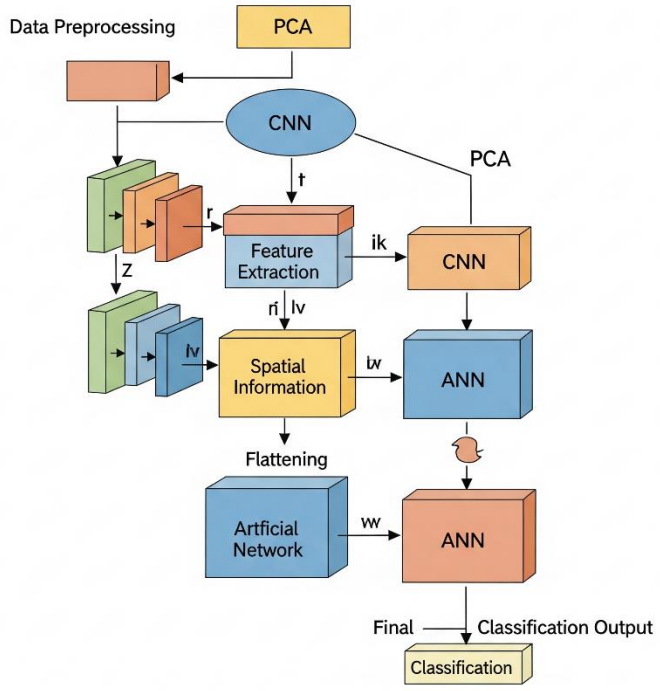

Figure 1: Data Preprocessing Method & Architecture Flow

Figure 1 explains data preprocessing method and how it passes to the hybrid model. The pseudo code outlines the architecture and workflow of a hybrid deep learning model that combines Artificial Neural

Network (ANN) and Convolutional Neural Network (CNN) components. The process starts with loading the dataset and converting class labels into a numerical format**,** which is necessary for machine learning algorithms to process categorical labels. To address class imbalance**,** the SMOTE (Synthetic Minority Over-sampling Technique) method is applied. SMOTE generates synthetic examples for minority classes, ensuring that each class has a balanced number of samples, which helps improve the model's learning and classification accuracy. Next, Principal Component Analysis (PCA) is used for feature selection and dimensionality reduction. PCA extracts the most significant features from the dataset, removing redundant or less useful ones. This helps in reducing the complexity of the model, speeding up training, and potentially improving performance. The data is then reshaped into a format suitable for input into a CNN, which typically requires a multi-dimensional input (such as 2D matrices or images). The hybrid model is then defined**,** consisting of both ANN layers (which are good at capturing general patterns) and CNN layers (which are effective for feature extraction, especially from spatial data). After defining the model architecture, the model is compiled**,** specifying the optimizer, loss function, and evaluation metrics. Finally, the model is trained on the preprocessed data**,** and once training is complete, it is evaluated**.** Various performance metrics such as accuracy, precision, recall, and F1-score are used to assess how well the model performs on the multiclass classification task.

H. Proposed ANN-CNN Model Architecture

Model: "sequential_2"

Layer (type)	Output Shape	Param #
conv1d_8 (Conv1D)	(None, 18, 50)	200
max_pooling1d_7 (MaxPooling1D)	(None, 9, 50)	0
conv1d_9 (Conv1D)	(None, 7, 45)	6,795
max_pooling1d_8 (MaxPooling1D)	(None, 3, 45)	0
flatten_2 (Flatten)	(None, 135)	0
dense_8 (Dense)	(None, 80)	10,880
dropout_6 (Dropout)	(None, 80)	0
dense_9 (Dense)	(None, 60)	4,860
dropout_7 (Dropout)	(None, 60)	0
dense_10 (Dense)	(None, 40)	2,440
dropout_8 (Dropout)	(None, 40)	0
dense_11 (Dense)	(None, 10)	410

Total params: 25,585 (99.94 KB)
Trainable params: 25,585 (99.94 KB)
Non-trainable params: 0 (0.00 B)

Figure 2: Proposed ANN-CNN Model Architecture

Figure 2 shows the proposed hybrid model, ANN-CNN. It contains the CNN layers, flatten layers, dropout layer, and fully connected dense layers ANN. It shows the layer type with the output shapes of the model.

Table 2. Comparison of Proposed Approach with Baseline Studies

No	Models	Accuracy
1	Proposed Approach	0.996%, 0.98%, 0.990%
2	ANN-GWO	0.9817%
3	CNN	0.98%

4	ANN	09017%

Table 2 shows a comparison between different existing approaches with our proposed approach. It contains a models accuracy comparison with the proposed architecture outcomes.

4. Results & Discussion

The experiment was conducted on a machine with an Intel Core i7 processor (8th generation) and 8 GB of RAM. Three multiclass datasets were used for testing the model's performance in this experiment. The development environment for the experiment was set up in PyCharm, a popular integrated development environment (IDE) for Python programming. The experiment also utilized Anaconda to manage the necessary libraries and dependencies.

For building and training the neural network, the TensorFlow and Keras libraries were employed, which are widely used for developing deep learning models. In addition, other Python libraries were used to visualize the model's performance, including plotting the confusion matrix, which helps in evaluating the accuracy and classification results of the model across different classes.

A. Microsoft Malware Dataset

The Microsoft malware dataset contains 10, 869 instances with ten malware families. The following performance by the hybrid model on this dataset is shown.

Table 3 Proposed Model Performance on Microsoft Malware Dataset

Model	Accuracy	Precision	Recall	F1-Score
Hybrid ANN-CNN	0.996%	0.981%	0.982%	0.983%
LSTM	0.90%	0.881%	0.883%	0.882%

Table 3 shows the proposed model's performance, including different types of metrics used in it. The model achieves 0.996% accuracy performance. There is a comparison of the deep learning model LSTM, after which it shows our hybrid model performs better than other deep learning models.

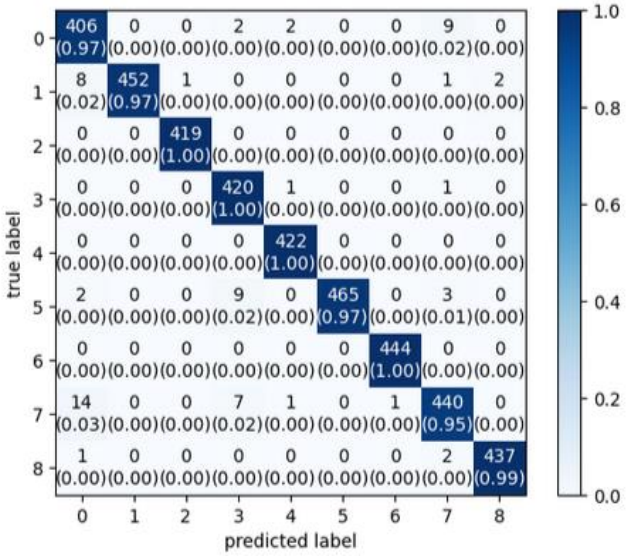

Figure 3 Confusion Matrix by Proposed Model on Microsoft Malware Dataset

Figure 3 shows per class predicted by proposed on Microsoft Malware Dataset. It contains ten malware families each class is correctly predicted by model.

B. WSN Intrusion Dataset

The wireless sensor network dataset contains 3 lac instances with five malicious families. The following performance by the proposed model in this dataset is shown as follows.

Table 4 Proposed Model Performance on Wireless Sensor Network Dataset

Model	Accuracy	Precision	Recall	F1-Score
Hybrid ANN-CNN	0.981%	0.951%	0.952%	0.953%
LSTM	0.971%	0.962%	0.963%	0.964%

Table 4 shows proposed model performance include different types of metrics used in it. The model achieves 0.981% accuracy performance.

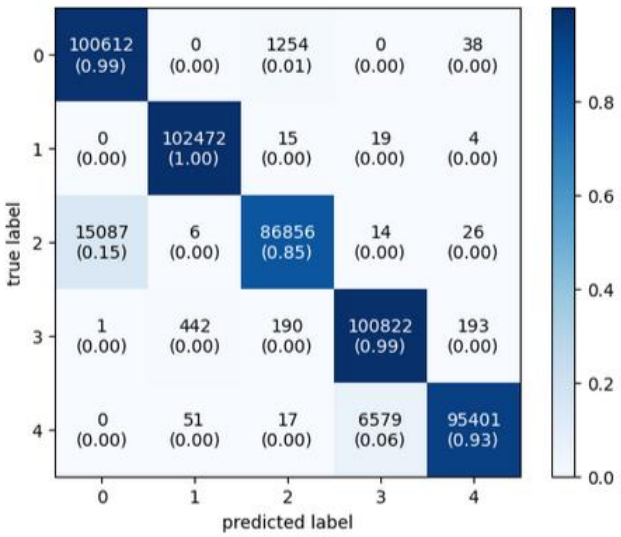

Figure 4: Confusion Matrix by Proposed Model in Wireless Sensor Network Dataset

Figure 4 presents the proposed model performance per class, show per class prediction performance.

I. Virus Malware Digit

The Virus Malware Digit dataset contains 10 malware families. The performance by the proposed model on the Virus Malware Dataset is shown as follows.

Table 5 Proposed Model Performance on Virus Digit Malware Dataset

Model	Accuracy	Precision	Recall	F1-Score
Hybrid ANN-CNN	0.990%	0.951%	0.952%	0.953%
LSTM	0.885%	0.891%	0.871%	0.861%

Table 5 shows the proposed model's performance, including different types of metrics used in it. The model achieves 0.990% accuracy performance. The comparison shows the proposed model performance and LSTM model performance, which states our approach is the best among other deep learning models in the classification of malware and intrusion.

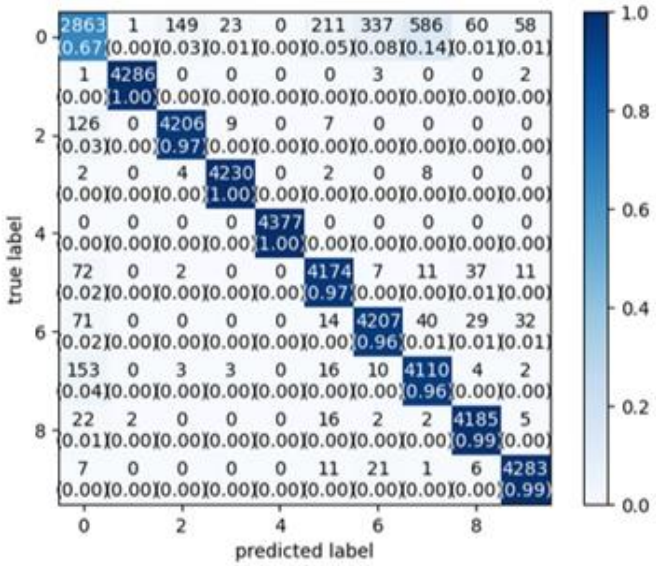

Figure 5: Confusion Matrix by Proposed Model in Virus Digit Malware Dataset

5. Discussion

From the experiments conducted, it can be observed that the proposed hybrid model (ANN-CNN) demonstrates excellent performance across all three tested datasets. The application of the class imbalance technique (SMOTE) played a key role in improving the model's effectiveness by ensuring that each class had a balanced number of samples. This prevented the model from making biased or misleading predictions, especially for minority classes.

Additionally, feature selection using PCA significantly contributed to the model's performance. By extracting only the most relevant features, PCA helped reduce training time and computational complexity, while still preserving important information needed for accurate classification.

For the Microsoft Malware dataset, which contains 10 malware classes, the model achieved an impressive accuracy of 99.6%. The confusion matrix further confirmed strong classification performance, with most samples correctly predicted.

In the second dataset, the WSN (Wireless Sensor Network) malware dataset, which includes 5 malware classes, the hybrid model also performed very well, achieving an accuracy of 98.5%. These results validate the effectiveness of the proposed hybrid model, the use of SMOTE for class balancing, and PCA for feature reduction.

6. Conclusion

In this paper, a hybrid ANN-CNN model is proposed and further enhanced by optimizing the model architecture, specifically by adjusting and improving the number and configuration of layers. To address the issue of class imbalance, the SMOTE (Synthetic Minority Over-sampling Technique) method is applied to the class labels, ensuring an equal distribution of samples across all classes. This helps the model learn more effectively and avoid bias toward the majority classes.

Additionally, Principal Component Analysis (PCA) is used for feature selection and dimensionality reduction. PCA selects the most informative features from the dataset, which helps reduce the dimensionality of the input data. This not only lowers the computational cost but also helps the model train more efficiently without sacrificing accuracy.

The enhanced hybrid model was evaluated on three multiclass datasets, achieving 99.6%, 98.1%, and 99.0% accuracy, respectively. These results demonstrate that the proposed approach offers strong and consistent performance, making it highly effective for multiclass classification tasks. In the practical deployment of the model, as this paper discusses, the model is hybrid, which means it requires highest fastest GPU-based system for this model that performs best. The model gives the best performance in different multiclass datasets. Big data or large datasets need to be trained in this model to determine how the model performs classification on a large dataset. Additionally, there is a risk of an overfitting problem in different types of datasets.

7. Future

For the coming days, advancements and more comprehensive evaluation, it is necessary to use bigger datasets that contain multiple categories to fully test the scalability and robustness of the proposed hybrid model. Injecting Particle Swarm Optimization (PSO) is also recommended. PSO is a heavy-duty optimization algorithm that can be used to fine-tune hyperparameters and further enhance the performance of the model by efficiently searching for optimal configurations.

Additionally, to increase the model's applicability and generalization across different domains, it is important to evaluate it on diverse types of datasets. These can include datasets with features such as:

- Windows API calls (to understand system behavior),
- PE (Portable Executable) headers (for binary structure analysis),
- Opcodes (to capture low-level code patterns in malware).

Using a joint of these dataset categories in a multiclass setting will provide a more thorough and realistic assessment of the hybrid ANN-CNN model's effectiveness across various malware detection and detection tasks.

References

[1] E. Venkata Pawan Kalyan, A. Purushottam Adarsh, S. Sai Likith Reddy, and P. Renjith, "Detection Of Malware Using CNN," *2022 2nd Int. Conf. Comput. Sci. Eng. Appl. ICCSEA 2022*, 2022, doi: 10.1109/ICCSEA54677.2022.9936225.

[2] M. Analysis, D. U. Machine, and L. Algorithms, "SS symmetry Malware Analysis and Detection Using Machine Learning Algorithms," 2022.

[3] J. Wang, S. Wang, and Y. Wang, "Malware Classification based on a Light-weight Architecture of CNN :

MalShuffleNet," pp. 2–5, 2022.

[4] S. I. Imtiaz, S. ur Rehman, A. R. Javed, Z. Jalil, X. Liu, and W. S. Alnumay, "DeepAMD: Detection and identification of Android malware using high-efficient Deep Artificial Neural Network," *Futur. Gener. Comput. Syst.*, vol. 115, pp. 844–856, 2021, doi: 10.1016/j.future.2020.10.008.

[5] W. Ma, C. Gou, and Y. Hou, "Research on Adaptive 1DCNN Network Intrusion Detection Technology Based on BSGM Mixed Sampling," *Sensors*, vol. 23, no. 13, 2023, doi: 10.3390/s23136206.

[6] H. Rana and Minhaj Ahmad Khan, "Detection of Malware Attacks using Artificial Neural Network," *VAWKUM Trans. Comput. Sci.*, vol. 11, no. 2, pp. 98–112, 2023, doi: 10.21015/vtcs.v11i2.1692.

[7] H. Rana, "Classification of Malicious Intrusion through ANN-CNN Sequential Classifier," *Int. J. Inf. Syst. Comput. Technol.*, vol. 3, no. 2, pp. 27–35, 2024, doi: 10.58325/ijisct.003.02.0088.

[8] M. Almousa and M. Anwar, "A URL-Based Social Semantic Attacks Detection With Character-Aware Language Model," *IEEE Access*, vol. 11, no. February, pp. 10654–10663, 2023, doi: 10.1109/ACCESS.2023.3241121.

[9] H. C. ALTUNAY and Z. ALBAYRAK, "Network Intrusion Detection Approach Based on Convolutional Neural Network," *Eur. J. Sci. Technol.*, no. 26, pp. 22–29, 2021, doi: 10.31590/ejosat.954966.

[10] K. Gupta, N. Jiwani, M. H. U. Sharif, R. Datta, and N. Afreen, "A Neural Network Approach For Malware Classification," *3rd IEEE 2022 Int. Conf. Comput. Commun. Intell. Syst. ICCCIS 2022*, pp. 681–684, 2022, doi: 10.1109/ICCCIS56430.2022.10037653.

[11] M. Kinkead, S. Millar, N. McLaughlin, and P. O'Kane, "Towards explainable cnns for android malware detection," *Procedia Comput. Sci.*, vol. 184, no. 2019, pp. 959–965, 2021, doi: 10.1016/j.procs.2021.03.118.

[12] N. Dwi, W. Cahyani, and H. H. Nuha, "Ransomware Detection on Bitcoin Transactions Using Artificial Neural Network Methods," pp. 669–673, 2021.

[13] Z. Fu, Y. Ding, and M. Godfrey, "An LSTM-Based Malware Detection Using Transfering Learning," *J. Cyber Secur.*, vol. 3, no. 1, pp. 11–28, 2021, doi: 10.32604/jcs.2021.016632.

[14] A. Sharma and U. Tyagi, "A Hybrid Approach of ANN-GWO Technique for Intrusion Detection," 2021.