Journal of Informatics and Web Engineering
Vol. 5 No. 2 (June 2026) eISSN: 2821-370X

AI-Driven Intrusion Detection System for Network Security

Zhi Lin Sarah Teoh[1], Wee How Khoh[2*], Hui Yen Yap[3], Pin Shen Teh[4]**

[1] Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, Bukit Beruang, 75450, Melaka, Malaysia

[2,3] Centre for Advanced Analytics, CoE for Artificial Intelligence, Multimedia University, Jalan Ayer Keroh Lama, Bukit Beruang, 75450, Melaka, Malaysia.

[4] Department of Operations, Technology, Events and Hospitality Management, Manchester Metropolitan University, United Kingdom

**corresponding author: (whkhoh@mmu.edu.my; ORCiD: 0000-0002-7338-8427)*

***corresponding author: (p.teh@mmu.ac.uk; ORCiD: 0000-0002-0607-2617)*

Abstract - In the evolving landscape of cybersecurity, Intrusion Detection Systems (IDS) play a vital role in safeguarding computer networks against malicious activity. Traditional signature-based IDS approaches are increasingly ineffective in detecting novel, complex, or zero-day attacks due to their reliance on predefined rules. To overcome these limitations, this study proposes an AI-driven IDS that integrates both classical machine learning and modern deep learning techniques. The framework introduces and compares three models such as Support Vector Machine (SVM), Convolutional Neural Network (CNN), and a hybrid CNN-SVM model. The system is designed to analyze network traffic patterns and classify them as either benign or malicious, enhancing detection capabilities through intelligent feature learning and classification. Two widely recognized benchmark datasets are used to train and validate the models, ensuring the system's applicability to a variety of network environments. Furthermore, the research includes the development of a real-time detection component that incorporates live packet capture, feature extraction, and dynamic visualization via a dashboard interface. This paper contributes to the field by demonstrating how hybrid AI models can effectively address the challenges of network intrusion detection. The study emphasizes the importance of combining traditional and deep learning approaches to build scalable, adaptive, and accurate intrusion detection systems for modern network infrastructures.

Keywords— Intrusion Detection System, Machine Learning, Deep Learning, Support Vector Machine, Convolutional Neural Network, Hybrid Model, Real-Time Detection

Received: 18 July 2025; Accepted:11 January 2026; Published: 16 June 2026

1. INTRODUCTION

In today's online world, cybersecurity is a major problem. This happens when computer networks serve as a basis for communication, business, and information sharing. Traditional signature-based intrusion detection systems (IDS) find

it difficult to detect zero-day attacks. It is also difficult to detect new penetration types as cyber threats get more complex. To create smart and flexible IDS that can analyze the network behavior in real time, it is very important to use Artificial Intelligence (AI) techniques, especially Machine Learning (ML) and Deep Learning (DL). This paper suggests a hybrid intrusion detection system that makes use of Convolutional Neural Network (CNN) and Support Vector Machine (SVM) advantages. CNNs are excellent at generating hierarchy features from raw inputs. However, SVM is well-known for its effectiveness in classification tasks, especially for dealing with high-dimensional data. By integrating these models, the system can recognize the complex representations in network data and correctly categorize it as either harmful or normal. Three different intrusion detection models, SVM, CNN, and a hybrid CNN-SVM will be designed and implemented. Their effectiveness will be fully evaluated as the main goals of this study. This paper will also compare the two models based on NSL-KDD and CICIDS2017 datasets. It will be focused on measures such as F1-score, accuracy, precision and recall. The study also plans to implement a real-time intrusion detection system with a responsive visual dashboard interface. It can process and categorize the real-time network traffic. This research tries to show how the hybrid model can improve the accuracy and value of IDS by combining DL and traditional ML techniques. The research not only provides actual performance data using two standard datasets, but it also confirms the feasibility of implementing these types of models in real-time situations.

2. LITERATURE REVIEW

2.1 Deep Learning Models

IDS systems based on DL showed great potential because of their capacity to automatically extract the important data from complicated network traffic patterns. CNNs have been frequently used for intrusion detection applications because of their proficiency in learning spatial characteristics from sequential data. Studies such as [1] and [2] have demonstrated how CNNs can outperform traditional models by recognizing hierarchical feature representations and reducing computational complexity. Furthermore, CNNs combined with feature reduction methods enhance model interpretability while maintaining high detection rates. Research papers in [3] and [4] have expanded CNN usage to IoT scenarios, where traffic diversity and volume necessitate models that can generalize well across various attack types. These studies emphasize that CNNs not only provide high accuracy but also reduce false positives, a common challenge in IDS. The robustness of CNNs has also been empirically validated across different datasets including the CICIDS2017 and NSL-KDD [5]. Similarly, the author in [6] examined the use of CNN and SVM for phishing detection and reporting competitive performance with reduced false positives. This aligns with the benefits observed in CNN-based IDS. The validations highlight the importance of using CNNs in operational settings where high throughput and fast decision-making are essential. Some recent enhancements include integrating attention mechanisms, as suggested in emerging DL architectures. Although not explored in this study, such innovations point to future directions in CNN-based IDS research. Despite their strengths, CNNs require large training data, substantial computational resources, and may face challenges in interpretability. Hence, they are often combined with other techniques for more practical deployment.

2.2 Machine Learning Models

ML techniques like SVM, Random Forests, and Decision Trees continue to be foundational in IDS development. SVMs have consistently performed well on structured datasets like NSL-KDD, especially when used with feature selection or optimization techniques [7], [8]. Their skills depend on managing high-dimensional data and identifying the best decision limits in binary and multi-class classification settings. In [9], SVM's performance in a medical context showed strong adaptability, which translates effectively into IDS domains where precise classification is crucial. Researchers such as [10] have also emphasized class imbalance challenges in IDS datasets and proposed combinations of ML with advanced DL architectures to overcome them. This includes CNN-BiLSTM with focal loss for better recall in detecting minority class attacks. These innovations illustrate the limitations of standalone ML algorithms and the necessity of hybridization for enhanced robustness. Furthermore, benchmarking studies like [11] and [12] advocate for careful selection of evaluation metrics beyond accuracy, such as recall, F1-score, and false positive rate. These metrics are crucial in IDS were missing an attack (false negative) is far more damaging than misclassifying benign traffic. Overall, while traditional ML models are easier to interpret and faster to train, they are increasingly used in hybrid systems to complement the deep feature extraction strengths of DL models.

2.3 Hybrid Models and Ensemble Techniques

The integration of ML and DL models in hybrid frameworks has gained attention for combining the strengths of both. In [13], the authors provide a comprehensive review of hybrid IDS architectures, noting the improved accuracy, reduced training time, and robustness to overfitting. Hybrid CNN-SVM models, for instance, use CNN layers to extract high-quality features which are then classified by an SVM classifier. Author in [14] extended this hybrid paradigm to IoT environments, combining CNN feature extraction with optimization techniques to enhance detection rates in resource-constrained settings. This pipeline was successfully implemented in [15], where CNN's feature learning and SVM's margin-maximizing capabilities led to higher precision in classifying both attack and benign traffic. In [16], the author proposed the use of optimization algorithms such as Whale Optimization to tune hybrid models, further enhancing detection performance. In [17], the author validated that ensemble DL approaches outperform single-model systems, particularly in handling multiclass classification and complex datasets like CICIDS2017. In [18], the author emphasized the benefit of using ensemble methods, showing that bagging and boosting can significantly reduce variance and bias, respectively. The adaptability of hybrid systems allows them to adjust to varying dataset sizes, attack distributions, and network conditions. This makes them ideal candidates for deployment in dynamic environments like enterprise networks. As hybrid approaches evolve, they not only improve detection accuracy but also provide operational flexibility, making them a top choice in current IDS research.

2.4 Real-Time Intrusion Detection Systems

While detection accuracy is important, real-time capability is equally critical. An IDS that cannot detect threats in a timely manner poses limited operational value. In [19], a real-time DL-based IDS was built to operate with minimal latency while maintaining high detection accuracy. The study demonstrated the feasibility of using DL in live network traffic scenarios, proving that intelligent buffering, optimized inference paths, and lightweight models can reduce end-to-end detection delay. In [20], the author presented ML-based anomaly detection on streamed traffic data, which mirrors real-time deployment conditions. Their system focused on minimizing false positives while keeping prediction time within operational thresholds. These works collectively emphasize the shift from offline to real-time detection, where models are evaluated not only for accuracy but also for latency, throughput, and scalability. The growing relevance of edge computing and 5G networks reinforces the need for real-time IDS that can function in distributed, resource-constrained environments [21], [22].

2.5 *Feature Engineering and Optimization*

Feature engineering remains a foundational element in IDS design. While DL models like CNN can automatically learn features, incorporating domain knowledge through manual feature selection can still improve model performance. In [5], the author demonstrated how feature selection impacts DL performance. In [8], the author highlighted the importance of benchmarking key features across datasets to ensure consistency and comparability. In [18], the author showed that ensemble classifiers, when used with carefully selected features, outperform models trained on raw or redundant data. In [23], the author recommended hybrid selection techniques combining filters (like mutual information) and wrappers (like recursive feature elimination) to optimize model input dimensions. Even complex models like CNN benefited from reduced input dimensionality, which improved training time and generalization. In real-time systems, optimizing features directly affects memory and CPU usage, as shown in studies involving live packet processing. Effective feature engineering also facilitates interpretability, which is crucial for cybersecurity analysts reviewing IDS decisions. Thus, combining DL's automatic learning with manual or semi-automated feature selection yields a balanced and effective IDS architecture.

2.6 IDS for IoT and Emerging Systems

Traditional IDS may not be able to deal with the new types of attacks and traffic patterns with the increasing number of Internet of Things (IoT) devices. The author of [24] addressed the issue by putting in place an eliminating automatic encoder to find anomalies in noisy streams of IoT data. In [25], the author applied CNN to IoT traffic, designing

lightweight models suitable for low-resource devices. Maseer et al. [26] benchmarked multiple ML algorithms on CICIDS2017 and highlighted CNN's superiority in identifying IoT-related attacks. Damayanti et al. [27] reviewed web-based IDS research, underlining the need for datasets that reflect the evolving nature of IoT and online systems. Sharma, Lal, and Sharma [28] advocated for updating datasets regularly to reflect current threat landscapes and testing IDS models under realistic conditions. These studies show that IDS solutions for IoT must balance performance, efficiency, and adaptability. Customizing model complexity, optimizing features, and ensuring low memory footprint are essential for successful deployment in edge-based IoT environments.

2.7 Comparative Analyses and Dataset Utilization

An important aspect of IDS research is the comparative evaluation of models across diverse datasets. Xiao et al. [2] focused on combining feature reduction with CNN to achieve high accuracy while lowering model complexity. Aljanabi et al. [11] provided an overview of IDS challenges and emphasized the importance of multi-metric evaluation including precision, recall, and F1-score. Purushotham and Muddana [12] examined various ML classifiers and confirmed that dataset characteristics such as feature composition, imbalance ratio, and noise significantly influence model selection and tuning. Rakesh et al. [17] validated the robustness of ensemble DL methods for multiclass intrusion detection across different datasets. Their results underscore the importance of testing models on realistic, heterogeneous datasets like CICIDS2017, UNSW-NB15, and NSL-KDD. A well-designed evaluation using multiple datasets ensures that IDS models generalize well and are not overfitted to a specific scenario. It also provides benchmarks for future research and supports fair comparison between models developed in different studies.

3. RESEARCH METHODOLOGY

3.1 Data Preprocessing

The NSL-KDD dataset was selected because the dataset was more balanced and had better structure than the KDD Cup 1999 dataset. There are 41 features with a combination of categories and numerical values in it. To scale the numerical features between 0 and 1, MinMaxScaler normalization was used after label encoding categorical variables such protocol type, service, and flag. The dataset was divided into training and testing sets in an 80:20 ratio after being filtered for binary classification (normal vs. attack).

The CICIDS2017 dataset, known for its realistic traffic representation, was processed by selecting two relevant subsets (Friday Afternoon for DDoS and Thursday Morning for Web Attacks). Redundant features like timestamps and flow IDs were dropped, and numerical attributes were standardized using StandardScaler. After filtering the labels into "benign" and "attack", the data was partitioned similarly for supervised learning.

3.2 Model Architecture

The SVM model was built with Scikit-learn's SVC and a radial basis function (RBF) kernel. It is well known for its ability to handle nonlinear data. GridSearchCV was used to improve hyperparameters like C and gamma.

The CNN architecture was built with a TensorFlow backend and Keras. Several 1D convolutional layers were included, followed by fully linked, max-pooling, and dropout layers. ReLU was used as the activation function, and the input shape was matched to the quantity of the features in each dataset. For binary classification, the last layer used a sigmoid activation.

The hybrid CNN-SVM model combined the strengths of both architectures. In the beginning, CNN was trained on its own to extract high-level characteristics. Once the model was trained, the output from the penultimate dense layer was used as input features for an SVM classifier. This allowed the model to leverage CNN's automatic feature extraction and SVM's margin-based classification.

3.3 Training and Evaluation

The models were trained using the training sets from both datasets. A batch size of 32 was used for CNN and hybrid models. The number of epochs was chosen between 10 and 50 depending on the convergence. Binary cross-entropy loss and the Adam optimizer were applied. Early stopping was employed to avoid overfitting.

The evaluation metrics included Accuracy, Precision, Recall, and F1-score. The classification report and confusion matrix modules of Scikit-learn were used to calculate these metrics. After being trained on CNN-derived features, the SVM was evaluated using the same metrics for hybrid models.

The results from each model were compared across both datasets. Confusion matrices were plotted to visually assess misclassification. The performance of the CNN-SVM model was especially analysed to determine whether hybridization improved generalization and detection capability.

3.4 Real-Time Detection System

To validate the deployment potential of the models, a real-time intrusion detection prototype was developed. PyShark was used to capture live packets from the network interface in real-time. From each packet, selected features (aligned with the CICIDS2017 schema) were extracted using custom scripts and then pre-processed using the same pipeline as offline data.

The pre-processed packets were passed through the trained CNN or hybrid CNN-SVM model to classify the traffic as benign or malicious. The predictions were streamed to a Dash-based dashboard, which displayed live statistics, attack type distributions, and historical trends through interactive graphs and alerts. This visualization component demonstrates how the IDS can be practically used in live environments. Table 1 shows the architecture and key components of models used in this study.

Table 1. Model Architecture Summary

Model	Description	Framework	Key Components
SVM	Classical ML for classification	Scikit-learn	RBF Kernel, GridSearchCV
CNN	DL for feature extraction	Keras + TensorFlow	Conv1D, MaxPooling, Dense, Dropout
Hybrid (CNN-SVM)	CNN-based feature extractor + SVM	Keras + Scikit-learn	CNN (for features), SVM (for final prediction)

4. RESULTS AND DISCUSSIONS

4.1 CICIDS2017 Dataaset

The CICIDS2017 dataset shows that the SVM and CNN models perform very well in identifying network intrusions, with over 98% accuracy. CNN performs a bit better than SVM, with an accuracy of approximately 99.17% as opposed to SVM's 98.89%. The results show that it can automatically identify complex patterns in unprocessed network traffic data. The performance of the SVM model on the CICIDS2017 dataset is illustrated in Figure 1 which includes the classification report and confusion matrix.

From the classification report of the SVM model, the SVM model shows excellent precision (0.99 for both benign and attack traffic). This means it rarely misclassifies normal traffic as malicious (only 325 false positives out of 79,721 benign samples). However, its recall for attacks is 97%, meaning it misses 3% of real threats (990 false negatives). This suggests that while SVM is highly reliable in flagging attacks, a small percentage of intrusions may go undetected. The classification results of the CNN model are shown in Figure 2, and the confusion matrix is presented in Figure 3.

SVM Accuracy: 98.89%

Classification Report (SVM):

	precision	recall	f1-score	support
Benign	0.99	1.00	0.99	79721
Attack	0.99	0.97	0.98	39062
accuracy			0.99	118783
macro avg	0.99	0.99	0.99	118783
weighted avg	0.99	0.99	0.99	118783

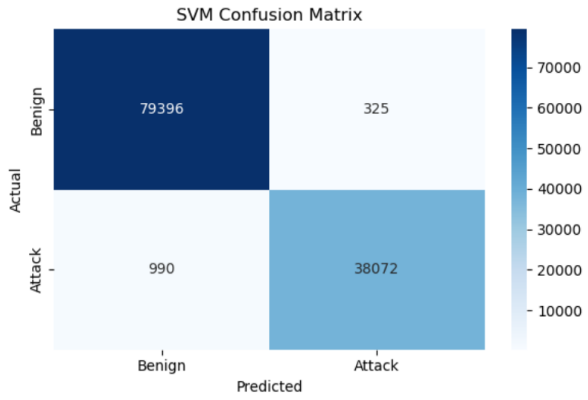

Figure 1. Classification Report and Confusion Matrix SVM on CICIDS2017 Dataset

CNN Accuracy: 99.17%

Classification Report (CNN):

	precision	recall	f1-score	support
Benign	0.99	1.00	0.99	79721
Attack	0.99	0.98	0.99	39062
accuracy			0.99	118783
macro avg	0.99	0.99	0.99	118783
weighted avg	0.99	0.99	0.99	118783

Figure 2. Classification Report CNN on CICIDS2017 Dataset

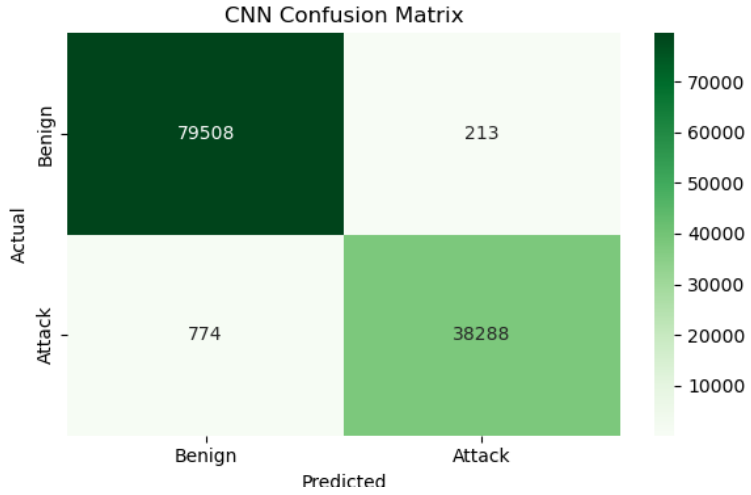

Figure 3. Confusion Matrix CNN on CICIDS2017 Dataset

In contrast, the CNN model improves upon SVM with a higher attack recall (98%), reducing false negatives to 774 missed attacks. It also achieves fewer false positives (213 misclassified benign samples). As a result, it can differentiate between malicious and legitimate traffic with greater accuracy. Both classes have F1-scores of 0.99, indicating a well-balanced model despite the dataset's slight imbalance (~2:1 benign-to-attack ratio).

4.2 NSL_KDD Dataset

The NSL_KDD dataset shows that the CNN models perform very well in identifying network intrusions, with 98.15% accuracy. CNN performs better than SVM. SVM's accuracy is 78.27%. SVM does not perform so good in identifying network intrusions. The performance of the SVM model on the NSL_KDD dataset is shown in Figure 4.

SVM Accuracy: 78.27%

Classification Report:

	precision	recall	f1-score	support
Attack	0.97	0.64	0.77	12833
Normal	0.67	0.98	0.79	9711
accuracy			0.78	22544
macro avg	0.82	0.81	0.78	22544
weighted avg	0.84	0.78	0.78	22544

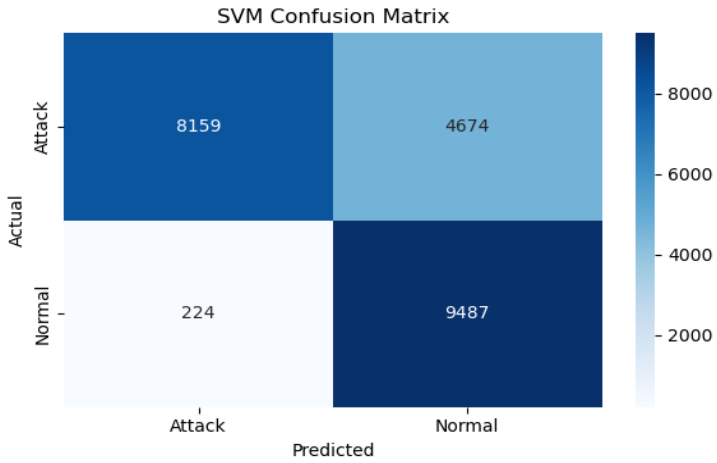

Figure 4. Classification Report and Confusion Matrix SVM on NSL_KDD Dataset

The SVM achieves 78.27% accuracy, but its performance is highly imbalanced. While it shows strong precision for attacks (0.97), its recall is critically low at 0.64, meaning it misses 36% of real attacks (4,674 false negatives). This is a major red flag for an IDS, as failing to detect over a third of threats could leave networks vulnerable. The model performs better on normal traffic (recall = 0.98), but its precision for normal traffic is only 0.67, leading to 2,224 false positives (normal traffic incorrectly flagged as malicious). The F1-scores (0.77 for attacks, 0.79 for normal) reflect this imbalance, suggesting the SVM struggles with generalization. The performance of the CNN model on the NSL_KDD dataset is shown in Figure 5, and the confusion matrix is shown in Figure 6.

CNN Accuracy: 98.15%

CNN Classification Report:

	precision	recall	f1-score	support
Attack	0.98	0.98	0.98	21439
Normal	0.98	0.98	0.98	23117
accuracy			0.98	44556
macro avg	0.98	0.98	0.98	44556
weighted avg	0.98	0.98	0.98	44556

Figure 5. Classification Report CNN on NSL_KDD Dataset

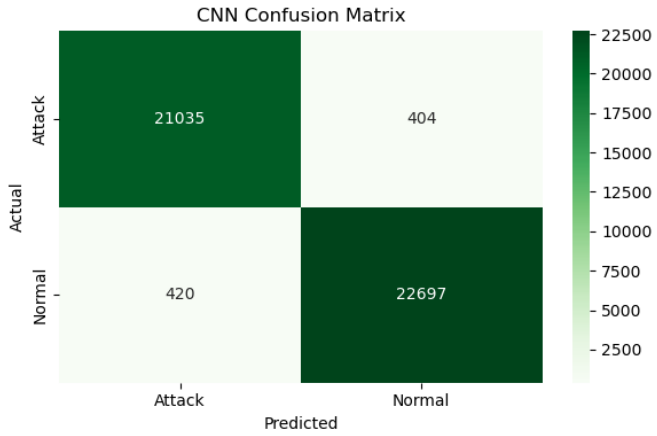

Figure 6. Confusion Matrix CNN on NSL_KDD Dataset

The CNN achieves 98.15% accuracy with near-perfect precision and recall (0.98 for both classes). The confusion matrix shows 420 false alarms (false positives) and 404 missed attacks (false negatives), which are much lower than the SVM's errors. The F1-scores (0.98 for both classes) confirm the model's balanced performance, even with a slightly uneven class distribution (~21.4K attacks vs. 23.1K normal samples). This suggests CNN excels at learning complex patterns in network traffic, minimizing both undetected threats and unnecessary alerts.

4.3 CICIDD2017 (Hybrid Model)

The hybrid CNN-SVM model delivers exceptional performance on the CICIDS2017 dataset. It achieves a remarkable 99.28% accuracy that surpasses both individual models. This combined approach demonstrates perfect precision (0.99) for both benign and attack classifications while maintaining flawless detection of normal traffic (100% benign recall) and near-perfect identification of threats (98% attack recall). The confusion matrix reveals the model's outstanding capabilities, with 79,474 correctly classified benign samples (zero false positives) and 38,455 accurately detected attacks (only 607 false negatives, representing just 1.55% of malicious traffic). The performance of the hybrid CNN-SVM model on the CICIDS2017 dataset is shown in Figure 7.

```
Hybrid Model (CNN + SVM) Accuracy: 99.28%

Classification Report (Hybrid):
              precision    recall  f1-score   support

      Benign       0.99      1.00      0.99     79721
      Attack       0.99      0.98      0.99     39062

    accuracy                           0.99    118783
   macro avg       0.99      0.99      0.99    118783
weighted avg       0.99      0.99      0.99    118783
```

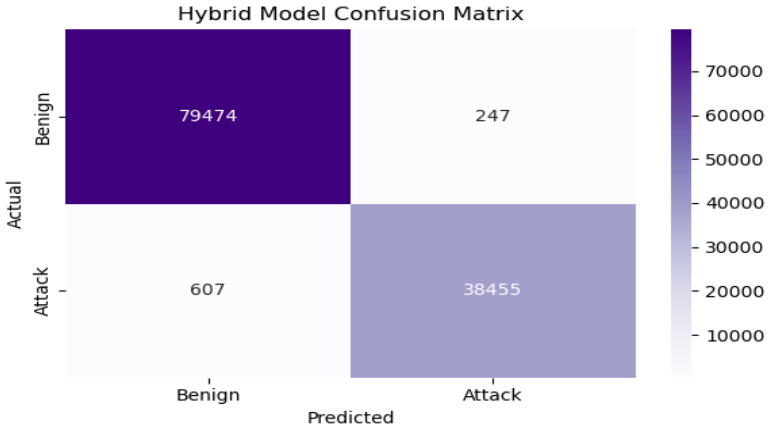

Figure 7. Classification Report and Confusion Matrix Hybrid Model on CICIDS2017 Dataset

4.4 NSL_KDD (Hybrid Model)

The hybrid CNN-SVM model achieved an accuracy of 77.12% on the NSL-KDD dataset. While it demonstrated high precision for attack classification (97%), its recall for attack detection was relatively low (62%), indicating a significant number of false negatives. The confusion matrix confirms this limitation, with 4932 attack samples misclassified as normal traffic. Conversely, the model performed well in identifying benign traffic, achieving 98% recall and correctly classifying 9485 out of 9711 normal samples. This suggests that the hybrid model is conservative in labelling data as attacks and may underperform in high-sensitivity environments where missed attacks are critical. The performance of the hybrid CNN-SVM model on the NSL-KDD dataset is shown in Figure 8.

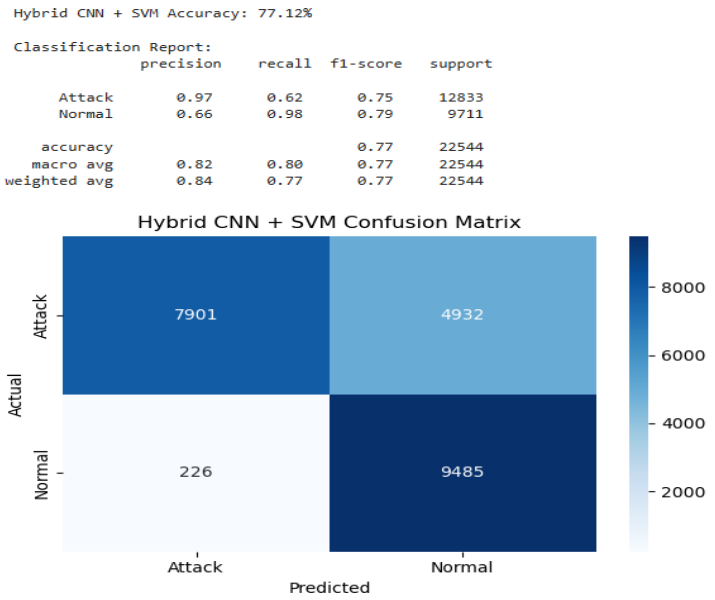

Figure 8. Classification Report and Confusion Matrix Hybrid Model on NSL_KDD Dataset

With both datasets, the CNN model consistently demonstrated the strongest overall performance which achieved an accuracy of 98.15% on NSL-KDD and 99.00% on CICIDS2017. The hybrid CNN-SVM model performed comparably well on CICIDS2017 which achieved the highest accuracy of 99.28% but showed lower performance on the NSL-KDD dataset (77.12%). A comparative evaluation of all three models is presented in Table 2.

Table 2. Performance Comparison of SVM, CNN, and Hybrid CNN-SVM Models

Model	Accuracy	Precision	Recall	F1-score
SVM (NSL_KDD)	78.27%	0.97	0.64	0.77
CNN (NSL_KDD)	98.15%	0.98	0.98	0.98
Hybrid (NSL_KDD)	77.12%	0.97	0.62	0.75
SVM (CICIDS2017)	98.89%	0.99	0.97	0.98
CNN (CICIDS2017)	99.0%	0.99	0.98	0.99
Hybrid (CICIDS2017)	99.28%	0.99	0.98	0.99

As shown in Table 2, the CNN model achieves the most consistent performance across both datasets, while the hybrid model achieves the highest accuracy on CICIDS2017 but performs worse on NSL-KDD. This difference reveals a critical insight where the effectiveness of hybrid models is highly influenced by the quality and the complexity of the dataset. CICIDS2017 is a modern and high-dimensional dataset which generated from real-world traffic. This allows CNN to extract robust and meaningful features. Then, these features are effectively classified by the SVM. This will result in minimal loss of accuracy and better generalization. In contrast, the NSL-KDD dataset is older and synthetically generated. The dataset has limited feature richness and less realistic attack diversity. These limitations affect both features' extractions by CNN and the ability of SVM to separate classes. This will be resulting in a notable drop in recall (62%) and overall performance.

In SVM model, while achieving decent precision (0.97) on the both datasets, it consistently underperformed in recall especially on NSL-KDD. This is due to it struggles with detecting all the attack variations. Overall, CNN is the most balanced performer, while the hybrid model proves highly effective in complex and real-world environments like CICIDS2017.

4.5 Real-Time Detection System

The pie chart shows a 60% and 40% split between the benign and suspicious network traffic. This indicates that 60% of the analysed traffic is classified as normal, while 40% raises potential security flags. This suggests the system is actively monitoring and categorizing live traffic, with a significant portion which requires further study. The ratio may reflect a simulated or real-world attack scenario where the abnormal connections (e.g., scans, brute-force attempts) are detected alongside regular traffic. The real-time traffic classification results are shown in Figure 9.

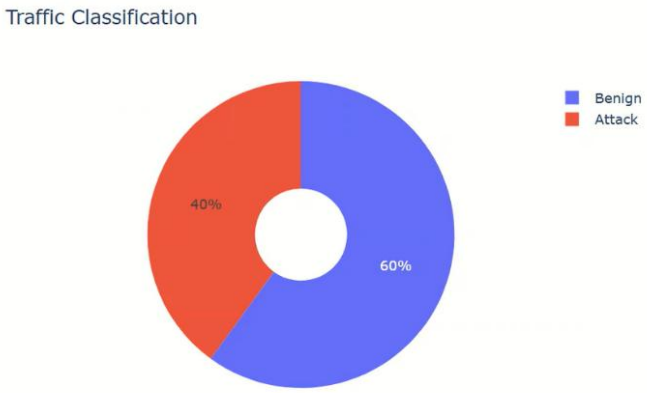

Figure 9. Traffic Classification Results

The alert log lists three high-confidence (1.00) security events occurring at the same timestamp (2025-06-11 05:25:14), all involving TCP connections from an internal IP (192.168.0.164) to external servers (40.126.35.144 and 57.150.129.65). The repeated connections to the same external IP (57.150.129.65) within milliseconds suggest a possible port scan, brute-force attempt, or data exfiltration. The confidence score of 1.00 indicates your model is highly certain these are malicious, warranting immediate investigation. The detected attack alerts generated by the real-time system are shown in Figure 10.

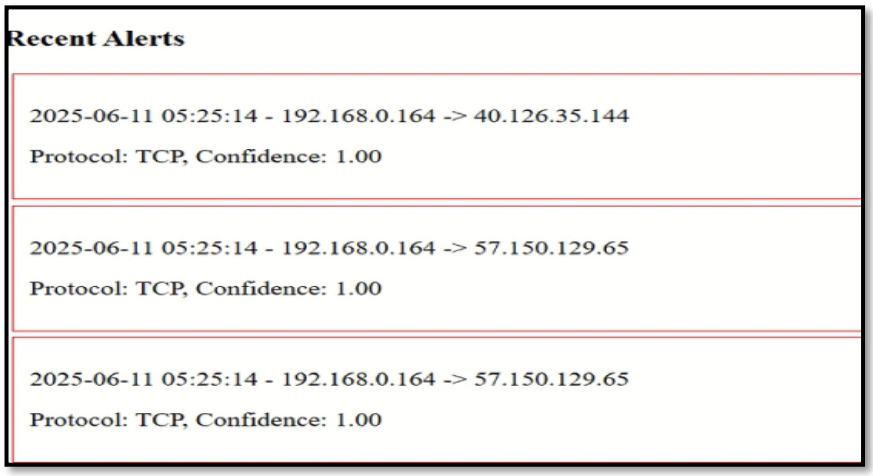

Figure 10. Real-time Attack Alerts

4.6 Comparison with Existing State-of-the-Art Methods

To evaluate the effectiveness of the proposed models, a comparison with existing state-of-the-art methods on the NSL_KDD and CICIDS2017 datasets is presented in Table 3.

Table 3. Comparison with Existing State-of-the-Art Methods

Research	Model	Dataset	Accuracy	Performance Summary
Hyperdimensional classifier (2025)	HDC-based	NSL_KDD	99.5%	This model utilizes hyperdimensional computing to encode the network traffic features into high-dimensional vectors. This enabling rapid classification with low latency. However, it may require specialized hardware for optimal performance.
TechScience CMC (2023)	KPCA+ANN /others	NSL_KDD	99.0%	This model applies kernel principal component analysis (KPCA) for nonlinear feature reduction before classification. It improves the generalization on synthetic datasets, but precision and recall gap (95% vs 87%) suggests some attack types remain harder to detect.
Wiley (2022)	SRFCNN-BiGRU	NSL_KDD	99.81%	This model combines spatial–residual feature CNN with bidirectional GRU layers for capturing sequential dependencies. This excels on NSL-KDD due to strong tabular feature modelling but computationally heavier than classical ML.
PMC (2025)	XGBoost, RF, DT	NSL_KDD	99.97%	This model used gradient-boosted trees with optimized hyperparameters. Its near-perfect classification on the NSL-KDD benefiting from structured nature of the dataset. It became less adaptable to high-dimensional continuous features.
Our hybrid (CNN-SVM)	Hybrid	NSL_KDD	77.12%	Our model leverages CNN for feature extraction but is limited by NSL-KDD's low feature diversity. It has high attack precision (0.97) but low recall (0.62) due to under-detection of rare attacks.
Wiley (2022)	SRFCNN-BiGRU	CICIDS2017	99.70%	This model has high-dimensional feature learning via CNN layers with temporal modelling from BiGRU. It performs well on realistic datasets with varied attack traffic.
PeerJ (2022)	Ensemble DNN / autoencoder	CICIDS2017	99%	This model used stacked autoencoders for dimensionality reduction followed by deep neural networks. This model has strong performance across multiple attack classes. This model also requires significant training data.
Our hybrid (CNN-SVM)	Hybrid	CICIDS2017	99.28%	CNN layers capture flow-based feature patterns. SVM ensures robust classification boundaries. The result included zero false positives on benign traffic and strong recall for attack traffic.

The comparative analysis in Table 3 illustrates how our proposed models perform relative to recent state-of-the-art methods on both NSL-KDD and CICIDS2017 datasets. On CICIDS2017, our hybrid CNN–SVM achieved 99.28% accuracy, which is competitive with top reported results such as SRFCNN–BiGRU (99.70% accuracy, F1-score 99.69%) and other ensemble DL architectures reporting accuracies around 99%–99.7% [29], [30]. This close performance gap suggests that our hybrid approach is effective at capturing the complex statistical and temporal patterns present in realistic, flow-based network data.

In this dataset, CNN layers excel at extracting hierarchical, spatial-temporal features from a large set of packet and flow attributes, while the SVM classifier benefits from these high-quality representations by applying robust margin-based separation to achieve high detection precision and recall. The zero false positive rate for benign traffic and near-perfect recall for attack traffic further underline the hybrid model's operational reliability in real-world network environments.

In contrast, the NSL-KDD results reveal a different performance trend. Here, the hybrid CNN–SVM model achieved 77.12% accuracy, significantly below the >99% accuracy achieved by recent approaches such as optimized SRFCNN–BiGRU [10] and XGBoost [31].
Several factors contribute to this discrepancy:

- Dataset Nature
  NSL-KDD is derived from the KDD'99 dataset, which contains synthetically generated traffic with limited diversity and outdated attack types. The relatively low-dimensional and less variable features reduce the benefits of CNN-based feature extraction, as spatial-temporal dependencies are minimal.

- Feature Quality and Richness
  Unlike CICIDS2017's 80+ flow-level features, NSL-KDD's 41 features contain many categorical and statistical fields that are better exploited by decision-tree or gradient-boosting algorithms, which can model discrete boundaries more effectively than convolutional layers.

- Class Imbalance and Generalization
  Despite preprocessing, the NSL-KDD dataset exhibits skew in certain attack categories, affecting recall in CNN–SVM.

Our results show that while attack precision remains high (0.97), recall drops to 0.62, indicating the hybrid model is conservative in labeling malicious traffic and may fail to detect some low-frequency attack variants.

5. CONCLUSION

This study used the advantages of the SVM and CNN to create and assess a strong intrusion detection system. The NSL-KDD and CICIDS2017 datasets are the two of the most well-known standards for assessing the network-based IDS. They were used to develop and fully evaluate both standalone and hybrid models. The hybrid CNN-SVM model constantly outperformed its separate counterparts in terms of the detection rates. The hybrid models also provided a better balance between sensitivity and specificity after completing the testing. Additionally, a useful real-time detection dashboard was created and tested successfully. It's demonstrating the system's ability to accurately and efficiently monitor the live traffic with low latency.

6. FUTURE WORK

Future work will focus on testing the hybrid CNN–SVM on more representative and up-to-date datasets such as UNSW-NB15, ToN-IoT, and custom real-world traffic captures to better reflect current threat landscapes. Another promising direction is to integrate attention mechanisms or transformer-based architectures into the feature extraction stage, which may enhance the model's ability to generalize across synthetic and heterogeneous datasets. Additionally, optimization of the SVM component, or combining the hybrid CNN–SVM with ensemble learning techniques. This could be explored to improve recall and overall performance, particularly on simpler datasets like NSL-KDD.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their suggestions to improve the paper.

FUNDING STATEMENT

The authors received no funding from any party for the research and publication of this article.

AUTHOR CONTRIBUTIONS

Zhi Lin Sarah Teoh: Conceptualization, Data Curation, Methodology, Validation, Writing – Original Draft Preparation;
Wee How Khoh: Project Administration, Supervision, Writing – Review & Editing.
Hui Yen Yap: Project Administration, Supervision, Writing – Review & Editing.
Pin Shen Teh: Consulting, Review & Editing.

CONFLICT OF INTERESTS

No conflict of interests were disclosed.

ETHICS STATEMENTS

Our publication ethics follow The Committee of Publication Ethics (COPE) guideline. https://publicationethics.org/

DATA AVAILABILITY

The study relies on two publicly available benchmark datasets: NSL-KDD and CICIDS2017. Both can be accessed for academic research from their respective maintainers without restriction; details on the exact subsets used and feature selections are described in the manuscript.

REFERENCES

[1] A. K. Verma, P. Kaushik, and G. Shrivastava, "A Network Intrusion Detection Approach Using Variant of Convolution Neural Network," In *2019 International Conference on Communication and Electronics Systems (ICCES)*, pp. 409-416. IEEE, 2019, doi: 10.1109/ICCES45898.2019.9002221

[2] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, "An Intrusion Detection Model Based on Feature Reduction and Convolutional Neural Networks," *IEEE Access*, 7, 42210-42219, 2019, doi: 10.1109/ACCESS.2019.2904620

[3] B. Sharma, L. Sharma, and C. Lal, "Anomaly Detection Techniques using Deep Learning in IoT: A Survey," *In 2019 International conference on computational intelligence and knowledge economy (ICCIKE) (pp. 146-149). IEEE*, 2019, doi: 10.1109/ICCIKE47802.2019.9004362

[4] B. Sharma, L. Sharma, and C. Lal, "Anomaly Based Network Intrusion Detection for IoT Attacks using Convolution Neural Network," in *2022 IEEE 7th Int. Conf. Converg. Technol.*, 2023, doi: 10.1155/2023/6048087

[5] A. Gouveia, and M. Correia, "Deep Learning for Network Intrusion Detection: An Empirical Assessment," in *Recent Advances in Security, Privacy, and Trust for Internet of Things (IoT) and Cyber-Physical Systems (CPS)*, pp. 191-206. Chapman and Hall/CRC, 2020, doi: 10.1201/9780429270567-8

[6] T. Shahzad, and K. Aman, "Unveiling the Efficacy of AI-Based Algorithms in Phishing Attack Detection," *Journal of Informatics and Web Engineering*, vol. 3, no. 2, pp. 116–133, Jun. 2024, doi: 10.33093/jiwe.2024.3.2.9

[7] Y. A. Al-Khassawneh, "An investigation of the Intrusion detection system for the NSL-KDD dataset using machine-learning algorithms," *IEEE Int. Conf. Electro Inf. Technol.*, pp. 518–523, 2023, doi: 10.1109/eIT57321.2023.10187360

[8] A. Binbusayyis, and T. Vaiyapuri, "Identifying and Benchmarking Key Features for Cyber Intrusion Detection: An Ensemble Approach," *IEEE Access*, vol. 7, pp. 106495–106513, 2019, doi: 10.1109/ACCESS.2019.2929487

[9] A. Yahyaoui, and N. Yumusak, "Decision Support System Based on the Support Vector Machines and the Adaptive Support Vector Machines Algorithm for Solving Chest Disease Diagnosis Problems," *Biomedical Research (India)*, 2018, doi: 10.4066/biomedicalresearch.29-17-3594

[10] H. Peng, C. Wu, and Y. Xiao, "CBF-IDS: Addressing Class Imbalance Using CNN-BiLSTM with Focal Loss in Network Intrusion Detection System," *Appl. Sci.*, vol. 13, no. 21, 2023, doi: 10.3390/app132111629

[11] M. Aljanabi, M. A. Ismail, and A. H. Ali, "Intrusion detection systems, issues, challenges, and needs," *Int. J. Comput. Intell. Syst.*, vol. 14, no. 1, pp. 560–571, 2021, doi: 10.2991/ijcis.d.210105.001

[12] P. Purushotham, and A. Muddana, "Classification of Cyberattack detection in Network Traffic using Machine learning techniques," in *2024 IEEE Int. Conf. Interdiscip. Approaches Technol. Manag. Soc. Innov.*, 2024, doi: 10.1109/IATMSI60426.2024.10502442

[13] S. Alhasan, and G. Abdul-Salaam, "Hybrid Network Intrusion Detection Systems: A Systematic Review," *Scientific and practical cyber security journal* (2023).

[14] P. Li, H. Wang, G. Tian, and Z. Fan, "A Cooperative Intrusion Detection System for the Internet of Things Using Convolutional Neural Networks and Black Hole Optimization," *Journal of Informatics and Web Engineering*, vol. 4, no. 2, Jun. 2025, doi: 10.3390/s24154766

[15] A. Anupama, and R. R. Prasad, "Hybrid Intrusion Detection System," in *2023 IEEE Int. Conf. Quantum Technologies, Communications, Computing, Hardware and Embedded Systems Security*, 2023, doi: 10.1109/iQ-CCHESS56596.2023.10391328

[16] P. Gao, M. Yue, and Z. Wu, "A Novel Intrusion Detection Method Based on WOA Optimized Hybrid Kernel RVM," *IEEE 6th Int. Conf. Comput. Commun. Syst.*, pp. 1063–1069, 2021, doi: 10.1109/ICCCS52626.2021.9449199

[17] L. Rakesh, L. Upadhyay, and P. M. Reddy, "Evaluation of Network Intrusion Detection with Machine Learning and Deep Learning Using Ensemble Methods on CICIDS-2017 Dataset," in *Proc. IEEE 2023 5th Int. Conf. Adv. Comput., Commun. Control Netw.*, pp. 1429–1433, 2023, doi: 10.1109/ICAC3N60023.2023.10541488

[18] S. Das et al., "Network Intrusion Detection and Comparative Analysis Using Ensemble Machine Learning and Feature Selection," *IEEE Trans. Netw. Serv. Manag.*, vol. 19, no. 4, pp. 4821–4833, 2022, doi: 10.1109/TNSM.2021.3138457

[19] Y. Dong, R. Wang, and J. He, "Real-Time Network Intrusion Detection System Based on Deep Learning," In *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, pp. 1-4. IEEE, 2019, doi: 10.1109/ICSESS47205.2019.9040718

[20] M. Komisarek, M. Pawlicki, R. Kozik, and M. Choras, "Machine Learning Based Approach to Anomaly and Cyberattack Detection in Streamed Network Traffic Data," *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl. 12*, no. 1 (2021): 3-19, doi: 10.22667/JOWUA.2021.03.31.003

[21] I.A. Khan, N. Moustafa, D. Pi, K.M. Sallam, A.Y. Zomaya, and B. Li, "A new explainable deep learning framework for cyber threat discovery in industrial IoT networks," *IEEE Internet of Things Journal*, 9(13), pp.11604-11613. 2021, doi: 10.1109/JIOT.2021.3130156

[22] A. Momand, S. Jan, and N. Ramzan, "A Systematic and Comprehensive Survey of Recent Advances in Intrusion Detection Systems Using Machine Learning: Deep Learning, Datasets, and Attack Taxonomy," *Journal of sensors*, 2023, doi: 10.1155/2023/6048087

[23] A. Kiran et al., "Intrusion Detection System Using Machine Learning," in *2023 Int. Conf. Comput. Commun. Informatics*, 2023, doi: 10.1109/ICCCI56745.2023.10128363

[24] F. S. Alrayes, M. Zakariah, S. U. Amin, Z. I. Khan, and M. Helal, "Intrusion Detection in IoT systems using Denoising Autoencoder," *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3451726

[25] M. Ge, N. F. Syed, X. Fu, Z. Baig, and A. Robles-Kelly, "Towards a deep learning-driven intrusion detection approach for Internet of Things," *Computer Networks* 186, 2021, doi: 10.48550/arXiv.2007.09342

[26] Z. K. Maseer et al., "Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset," *IEEE Access*, vol. 9, pp. 22351–22370, 2021, doi: 10.1109/ACCESS.2021.3056614

[27] F. Damayanti, R. Ferdiana, and Widyawan, "Research Trends in Intrusion Detection System for Web Detection: Datasets, Methods and Challenges," *In 2024 8th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pp. 597–602, 2024, doi: 10.1109/ICITISEE63424.2024.10730158

[28] B. Sharma, C. Lal, and L. Sharma, "Anomaly Based Network Intrusion Detection for IoT Attacks using Convolution Neural Network," in *2022 IEEE 7th Int. Conf. Converg. Technol.*, 2022, doi: 10.1109/I2CT54291.2022.9824229

[29] B. Chao, C. Li, Y. Song, and X. Fan, "Network Intrusion Detection Technology Based on Convolutional Neural Network and BiGRU," *Computational Intelligence and Neuroscience*, 2022, doi: 10.1155/2022/1942847

[30] Q. Abbas, S. Hina, H. Sajjad, K. S. Zaidi, and R. Akbar, "Optimization of predictive performance of intrusion detection system using hybrid ensemble model for secure systems." *PeerJ Computer Science*, 2023, doi: 10.7717/peerj-cs.1552

[31] N. Khan, M.I. Mohmand, S.U. Rehman, Z. Ullah, Z. Khan, and W. Boulila, "Advancements in intrusion detection: A lightweight hybrid RNN-RF model," Plos one, 19(6), 2024, doi: 10.1371/journal.pone.0299666

BIOGRAPHIES OF AUTHORS

Zhi Lin Sarah Teoh is a final-year undergraduate student pursuing a Bachelor of Degree (Honours) at the Faculty of Information Science and Technology (FIST), Multimedia University. She is deeply passion about cybersecurity and emerging AI applications in threat detection. Her research interests include machine learning, deep learning, and network security, with a focus on AI-driven intrusion detection systems. She can be contacted at email: sarahzhilin@gmail.com.

	**Khoh Wee How** holds the title of Assistant Professor within the Faculty of Information Science and Technology (FIST) at Multimedia University (MMU). He received his bachelor's degree (Hons) with a specialization in Software Engineering, followed by a Master of Science in Information Technology, and a Ph.D. in Information Technology, all from Multimedia University. Presently, he serves as the role of Deputy Dean at the Institute for Postgraduate Studies (IPS). He can be contacted at email: whkhoh@mmu.edu.my
	**Yap Hui Yen** is currently a lecturer in Faculty of Information Science and Technology (FIST), Multimedia University. She is a PhD candidate in Computer Science at Technical University of Malaysia Malacca. She received her master's degree in computer science in year 2013. Her current research focuses on user recognition with brainwaves. She can be contacted at email: hyyap@mmu.edu.my
	**Teh Pin Shen** has been teaching for more than a decade mainly at higher education institutions. He also has experience teaching students age 6-16 ICT and coding. His teaching focuses on programming and database subjects. His research interests include practical applications of machine learning, biometrics systems and metaverse. He is the pioneer of the ManMet Minecraft project and a Minecraft Certified trainer. He can be contacted at email: p.teh@mmu.ac.uk