IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 November 2024, accepted 30 December 2024, date of publication 10 January 2025, date of current version 21 January 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3528214

RESEARCH ARTICLE

Transferability Evaluation in Wi-Fi Intrusion Detection Systems Through Machine Learning and Deep Learning Approaches

SAUD YONBAWI[1], ADIL AFZAL[2], MUHAMMAD YASIR[3], MUHAMMAD RIZWAN[4], AND NATALIA KRYVINSKA[5]

[1]Department of Software Engineering, College of Computer Science and Engineering, University of Jeddah, Jeddah 23218, Saudi Arabia
[2]XeroAI, Lahore, Punjab 54890, Pakistan
[3]Department of Computer Science, University of Engineering and Technology Lahore, Lahore, Punjab 39161, Pakistan
[4]College of Science and Engineering, University of Derby, DE22 1GB Derby, U.K.
[5]Department of Information Management and Business Systems, Faculty of Management, Comenius University Bratislava, 820 05 Bratislava, Slovakia

Corresponding author: Adil Afzal (adilafzalansari@gmail.com)

This work was supported by the University of Jeddah, Jeddah, Saudi Arabia, under Grant UJ-22-DR-41.

ABSTRACT Intrusion Detection System (IDS) plays a pivotal role in safeguarding network security. The efficacy of these systems is rigorously assessed through established metrics including precision, recall, F1 score, and AUC score. When subjected to rigorous testing on well-known datasets like AWID and AWID3, individual IDS models consistently deliver exceptional performances, boasting F1 scores ranging from 0.98 to 1 and AUC scores spanning 0.97 to 0.99. However, the true challenge surfaces when the objective is to extend the transferability of these high-performing models to entirely novel, unseen datasets. This endeavor unravels a diverse performance landscape, demonstrating that the outstanding performance observed on a particular dataset doesn't guarantee the transferability of features across dissimilar datasets nestled within different network environments. In order to evaluate the feature transferability, we turn to AWID and AWID3 datasets as the main distinction between AWID (potentially referring to AWID2) and AWID3 lies in their specific focuses and contexts within the field of Wi-Fi intrusion detection. Although both datasets are centered on the general goal of detecting Wi-Fi intrusions, AWID3 has been carefully designed to meet the specific needs of corporate Wi-Fi applications. A comprehensive evaluation involving Multilayer Perceptron(MLP), and Convolutional Neural Networks (CNN) models has been executed, uncovering that CNN conspicuously outshines the MLP model.

INDEX TERMS Transferability assessment, performance evaluation, intrusion detection system (IDS), deep learning, wireless security.

I. INTRODUCTION

The worldwide cyber security environment has faced rising threats in recent years. Cyber fraudsters took advantage of misaligned networks as firms migrated to remote work settings during the epidemic. Malware assaults climbed 358 percent in 2020 compared to 2019 [1]. In 2020, cyber-attacks were anticipated to be the sixth most serious concern, and are now considered to be the new norm in the private as well as the public arenas. In 2023, this risky industry will continue to grow, with IoT cyberattacks alone expected to triple by 2025. Cyber-attack is a hostile attempt by one or more attackers to exploit vulnerabilities in a network in order to obtain unauthorized access, steal sensitive information, or disrupt usual network operations. Globally, cyberattacks surged by 38 percent in 2022 versus 2021 [2]. In an era marked by the pervasive integration of wireless communication technologies into our daily lives, the security of Wi-Fi networks has become paramount. With the exponential growth of connected devices and the continuous evolution of network threats, IDS plays a pivotal role in safeguarding the integrity and confidentiality

The associate editor coordinating the review of this manuscript and approving it for publication was Li Zhang.

of data transmitted over Wi-Fi networks. Traditional IDS techniques have shown remarkable efficacy in detecting known attacks; however, they often struggle to adapt to the dynamic and ever-evolving landscape of wireless network intrusions [3], [4]. In the realm of wireless communication, the security of Wi-Fi networks remains a paramount concern, spanning across a wide spectrum of applications, from small-scale home networks to large-scale industrial and enterprise systems. The effective detection of intrusions and anomalies in such networks is indispensable to ensure the confidentiality and integrity of transmitted data. Traditional IDS have exhibited noteworthy efficacy in detecting known threats within specific network contexts. However, their transferability to the diverse and dynamic landscape of Wi-Fi network environments remains a persistent challenge [5]. Feature transferability refers to the ability of features learned in one domain or dataset to be successfully utilized or transferred to a distinct domain or dataset. It refers to the concept that features extracted or acquired from one r dataset can potentially be reused or transferred to another dataset, even if both are not explicitly the same, in the context of machine learning and deep learning [6]. The idea of feature transferability is especially important in situations where getting labeled data for a new job is prohibitively expensive, time-consuming, or impossible. Researchers explore the prospect of utilizing information learned from a pre-trained model on a related job rather than beginning from scratch and training a new model from the ground up. This is also known as transfer learning or domain adaptation. The purpose of testing feature transferability is to discover the strongest and most advantageous features that can be efficiently transferred or generalized across various network environments, especially between a general network environment and an enterprise network environment. This transferability is vital for developing IDS models that can effectively recognize intrusion patterns in a variety of network scenarios, ranging from small-scale residential networks to large-scale industrial and enterprise networks.

The main objectives of this research are to:

- Investigate the transferability of dataset features across diverse network environments.
- Analyze the factors influencing the transferability of features and their impact on model performance.
- Reduce the False positive rate to develop a model that has a better prediction rate than the existing models.

II. LITERATURE SURVEY

In the ever-expanding landscape of wireless communication, Wi-Fi IDS stand as sentinels guarding the integrity of networks against an array of threats. The efficacy of these systems relies profoundly on their ability to detect intrusions and anomalies within the intricate tapestry of Wi-Fi network traffic. With the advent of Deep Learning, the promise of more robust and adaptable Wi-Fi IDS has come to the fore. A critical aspect of harnessing the potential of Deep Learning in this context is the evaluation of the transferability of dataset features across different network environments. Particle Swarm Optimization (PSO) with Deep Belief Network (DBN) is used to detect network intrusions in NSL-KDD [7]. Although 82.3 percent accuracy is achieved on the test dataset, it takes longer training time and detection time which affects the fitness function of the model due to the large size of hidden layers and high computational cost. [H] In order to enhance network security through intrusion detection, a Deep Residual Convolutional Neural Network (DCRNN) is proposed in this work [7].The Improved Gazelle Optimisation Algorithm (IGOA) is used to optimise the DCRNN. In order to exclude unnecessary features from the network data utilized for intrusion detection, feature selection is done. Using the Novel Binary Grasshopper Optimisation Algorithm (NBGOA), the key features are selected. The CIC-IDS-2017, Cicddos2019, and UNSW-NB-15 datasets are used in the experiments. The experimental results showed that, in terms of false alarm rate and processing time, the proposed model outperformed state-of-the-art methods. Deep Neural Network (DNN) is applied on several benchmark datasets NSL KDD, KYOTO, CIC IDS2017 for intrusion detection [8]. Despite high accuracies for every dataset, complex DNN architectures are not applied because of high computational costs and longer training/prediction time. Both NSL KDD and KYOTO are out dated datasets. The work in [9] proposed a hybrid technique to detect intrusions based on feature selection and classification using UNB ISCX 2012 and CIC IDS2018 datasets in the Apache Spark environment. Stacked Auto Encoder (SAE) performed feature selection and SVM for intrusion detection. Results demonstrated that 90.2 percent accuracy has been achieved with reduced training time. One disadvantage is that both of these datasets are highly imbalanced. One shortcoming is that UNB ISCX 2012 is an outdated dataset and consists of 6 attacks only. In this work [10], multiple supervised techniques such as ANN, DT, RF and unsupervised techniques such as k-means, self-organizing map (SOM), and expectation maximization (EM) algorithms are applied on CIC IDS2017. Some algorithms demonstrated high accuracy while others such as SOM and EM failed to detect attacks. The main drawback of this paper is that the dataset is highly imbalanced and has multiple features up to 80 but dimensionality reduction or feature selection is not considered. Furthermore, ROC and FAR scores are not given. In this work [11], the author investigated the application of machine learning within the context of two public datasets, with a specific focus on Denial-of-Service (DoS) attacks. The author trained three promising IDS — namely decision tree, random forest, and deep neural network— using flow records from the CIC-IDS-2017 dataset, encompassing both legitimate traffic and DoS attacks. Initially, they assessed the performance of each IDS model on separate data from the CIC-IDS-2017 dataset, achieving recall scores ranging from 0.97 to 0.99. This research proposed an effective machine learning-based automated intrusion detection system [12]. First, features

TABLE 1. Critical summary of literature.

Reference	Year	Problem	Technique	Dataset	Result	Gap	Transfer-ability
[13]	2022	Krack detection using ML	LightGBM, XGBoost, Catboost	AWID3	Acc = 87.12%	High false positive rate	No
[14]	2023	To detect Wi-Fi network attacks using reinforcement method	EBDM-DNN	AWID	F1-Score = 99.25%	Test evaluation on same kind of network traffic	No
[15]	2020	Feature reduction for wireless IDS	FFDNN	UNSW_NB15, AWID	Acc = 77.17%, 99.77%	Proposed methods were dataset-specific, and their performance downgraded on different Wi-Fi network configurations	No
[16]	(2023)	To detect network attacks using reinforcement method	AE-SAC	AWID	F1-Score= 98.9%	Test evaluation on same kind of network traffic	No
[17]	2020	Two stage wireless IDS	RF, NB, SHAP, ET, XGBoost, Bagging, LightGBM	AWID	Acc = 99.99%	The study lacked evaluation on different datasets, limiting its practical application	No
[18]	2022	Wireless IDS for 5g networks	DT, kNN, Decision Jungle, Decision Forest, Neural Networks	AWID3	Acc = 99%	Proposed method showed high accuracy but failed to generalize on new datasets properly	No
[40]	2022	Wireless IDS	Machine learning models	AWID, AWID3	F1-Score = 95.93%	F1-Score should be improved	Yes
[41]	2021	Transferability in federated setup	DNN	CIC-IDS-2017	accuracy = 70%	margin for improvement	Yes
[42]	2024	Transferability in adversarial environment	DNN,CNN LSTM	Survival	F1-Score = 94%	need of improved F1-Score	Yes
[43]	2024	Transferability evaluation	ADNN	Customised dataset	RMSE = 4.2%	need of improved F1-Score	Yes
[44]	2024	Transferability evaluation	AE	Customised dataset	acc = 90%	need of improved F1-Score	Yes

are extracted by the use of Modified Singular Value Decomposition (M-SvD). From the supplied data, M-SvD extracts important information such as basic, content, and traffic features. The Opposition-based Northern Goshawk Optimisation algorithm (ONgO) is then used to optimize these extracted features in order to increase their efficacy. Following feature selection, a hybrid machine learning model known as the Mud Ring assisted multilayer support vector machine (M-MultiSVM) is used to classify attacks. With the CSE-CIC-IDS 2018 dataset, the system obtained 99.89% accuracy, and with the UNSW-NB15 dataset, it scored 97.535% accuracy, according to evaluation metrics. Another research proposed GSAFS-OQNN model [11], a novel approach to intrusion detection and classification, is presented in this work. It selects the most significant features by applying the GSAFS model. Next, to find intrusions,

a Quantum Neural Network (QNN) is employed. The Sandpiper Optimisation (SPO) method is used to modify the QNN model's parameters. Standard Intrusion Detection System (IDS) datasets are used to evaluate the GSAFS-OQNN model. The GSAFS-OQNN method outperforms other current approaches, according to the results. To identify KRACK assaults, three gradient boosting techniques are used: LightGBM, XGBoost, and CatBoost, which passively monitor numerous wireless channels [12]. This technique, known as kTRACKER, achieves an accuracy of approximately 93.39% with a false positive rate of 5.08%. The novel Ensemble Binary Detection Model (EBDM) model proposed in another research [13] aims to transform the way we identify various kinds of threats. EBDM employs distinct models for every kind of attack, as opposed to attempting to identify multiple attack types simultaneously. Then, in comparison to conventional techniques, it mixes the best outcomes from different models to increase accuracy. The results demonstrate that, in comparison to previous techniques, EBDM is more effective at identifying three types of wireless attacks in the AWID dataset. The work [14] proposed a new method for detecting wireless intrusions that employs a Wrapper Based Feature Extraction Unit (WFEU) and a Feed-Forward Deep Neural Network (FFDNN). The system was evaluated using two separate intrusion detection datasets and compared to typical machine learning algorithms. The FFDNN with WFEU outperformed other approaches, attaining accuracies of 87.10% and 77.16% for binary and multiclass assaults, respectively, using a feature vector of 22 attributes in UNSWNB15 and a reduced feature vector of 26 attributes. Accuracy of 99.66% and 99.77% for binary and multiclass attacks, respectively, were achieved using a reduced feature vector of 26 characteristics using the AWID dataset. In this research, a new approach is proposed for detecting network intrusions on computers: Autoencoder-Soft Actor-Critic (AE-SAC) [15]. It uses reinforcement learning to learn from the environment. By competing against itself, the proposed approach is trained to make the appropriate decisions by giving different incentives for different types of attacks. AE-SAC exhibited sustainable results with an accuracy of 84.15% and an F1-score of 83.97% on the AWID dataset, as well as accuracy and a f1-score above 98.9%. Another suggested approach [16] consists of two consecutive steps that collaborate to determine if network records are normal or belong to specific attack classes. Machine learning models such as Random Forest (RF), Naïve Bayes (NB), SHAP, Extra Trees (ET), XGBoost, Bagging, and LightGBM are used in each stage to analyze the AWID dataset. Despite having fewer features, this two-stage Wireless Network Intrusion Detection System (WNIDS) achieves an astounding 99.42% accuracy for multi-class categorization. Another research presented a smart intrusion detection system that detects Internet of Things-based threats using a deep learning algorithm [17]. The approach ensured network security while also supporting IoT connectivity standards. The proposed approach recognizes global intruders and detects them using a neural network. The autoencoder model is outperformed, with 99.76% accuracy, illustrating the efficacy of user-centric cybersecurity solutions in 5G networks. Similarly, another research study [18] predicted binary and multiclass classifications using 13 and 76 feature sets, respectively. Deep Learning techniques, such as CNN, RNN-LSTM, DNN (3), and DNN (5), achieved accuracies ranging from 88% to 97%, whereas Machine Learning approaches achieved 88% to 98% accuracy. Another study [19] compared intrusion detection methods across numerous nominal, numeric, and binary categories. Feature selection strategies are used to enhance performance. A machine learning-based technique is applied, with boosted decision trees demonstrating higher performance and Logistic Regression achieving 99% accuracy. The proposed WiFi Intrusion Detection System detects WiFi attacks that traditional approaches miss using a lightweight machine learning model and optimized feature selection [20]. It employs a Light Gradient Boosting Machine and Gradient-based One Side Sampling to improve accuracy, precision, recall, and F1 score while shortening training and testing timeframes. The evil twin attack on AWID3 is explored in another study using data science methods [21]. The optimized LightGBM model achieved a False Positive Rate (FPR) of 0.00602 and 0.00898, indicating the potential for an effective Intrusion Detection System. A novel WiFi intrusion detection framework is presented in the study [22], utilizing artificial neuron training and the bio-inspired optimization algorithm, Harris Hawks optimization (N-HHO). The framework outperforms other models, suggesting its potential for enhancing network security on the AWID dataset. A reinforcement-learning-based intrusion detection approach is described in the research [23], utilizing the adaptive sample distribution with the dual-experience replay to address concerns regarding uneven data distribution and enhance classification accuracy in minority attacks. Another paper [24]proposed a wrapper-based feature extraction strategy that used the Extra Trees classifier on the NSL-KDD intrusion detection dataset, resulting in improved performance with a validation accuracy of 99.35%, an F-Measure of 99.67%, and a test accuracy of 88.42%. The study [25] presented an IDS that used multiple deep reinforcement learning agents to identify and categorize new and sophisticated attacks, with higher accuracy and lower False Positive Rate (FPR) than existing systems, as tested on three benchmark datasets. In this paper [26], the top percentile and recursive features were selected using the second percentile methodology, and recursive feature removal was performed using data from NSL-KDD, CIC-IDS-2017, and AWID, resulting in reduced computing time and complexity. In addition, a sparse autoencoder with swish-PReLU activation was used to categorize traffic kinds in datasets. Experimental results show a 4.77% improvement in classification accuracy.Similarly, another research [27] introduced an intrusion detection technique for WiFi networks that incorporated convolutional neural networks. The Dropout approach minimized training time and overfitting risk.

The experimental findings demonstrated 99% accuracy on AWID dataset. This research [28] presented two approaches for detecting intrusions in WiFi datasets, an incremental semisupervised graph-based clustering methodology and a quick outlier identification method, and demonstrated their usefulness in network security. Experiments on datasets from AWID and UCI demonstrated the high performance of the proposed strategies. The study [29] found that a weight-based machine learning model beats filters in feature selection such that 99.72% F1 score on AWID and that combining this method with a basic classifier increases performance. In this study [30], two-dimensional data cleansing is used, with 18 key qualities chosen from a larger collection of 154. A support vector machine (SVM) is then used to detect attacks using the cleaned data. Experimental analysis showed that flooding attacks are correctly identified 89.18% of the time, injection attacks 87.34% of the time, and normal traffic 99.88% of the time. The study implemented a multi-step feature selection approach [31], beginning with 32 features based on manual selection and previous research. The Correlation feature set (CFS) method was combined with the Harmony Search technique, yielding five features. The CFS evaluator was combined with the Classification with AntSearch method, resulting in 7 features. The CFS algorithm was then combined with BeeSearch, yielding ten characteristics. The experimental evaluation utilized AWID with seven machine learning classifiers: AdaBoost, Random Forest, Random Tree, J48, logit Boost, Multi-Layer Perceptron, and ZeroR. The system obtained a maximum accuracy of 99.64% using the Random Forest method with 32 features, and 99.53% using logit Boost with five features. This paper [32] presents a hybrid neural network (HNN) model that combines multi-feature correlation with temporal-spatial analysis. It uses a contribution-based feature selection method, CNN and LSTM for temporal-spatial information, and a Deep Neural Network (DNN) for intrusion detection. Another paper [33] presented a deep autoencoder dense neural network model that detects vulnerabilities in 5G and IoT networks using the AWID dataset with 99.9% accuracy rate. In this work [34], two feature selection techniques were used to detect injection attacks: constant removal and recursive feature elimination. The effectiveness of these techniques was evaluated using three classifiers Random Forest, SVM and Decision Tree. Experimental results utilizing the AWID dataset showed that the Decision Tree classifier performed the best in detecting injection assaults. When using the Decision Tree classifier with only eight features, the proposed technique for detecting injection attacks achieved 99% accuracy, 95% precision, and 90% F1 score. Another study [35]applied two environments to collect training and testing data for zero-day attacks. The data was generated to resemble real-time attacks, and features were ranked using an explainable AI(XAI) model. A time series generative adversarial network (TGAN) was used to analyze the top 12 features. The training data was integrated with the AWID dataset, and a hybrid deep learning model CNN-LSTM was used. The combined dataset performed better, with 93.53% accuracy using only the AWID dataset. The study [36] proposed a Fuzzy C-Means (FCM) feature selection method for wireless intrusion detection. It computes the difference between normal and attack center points and uses this information to choose features. The method has been evaluated on the AWID dataset, displaying high accuracy in binary classification of flooding, impersonation, and injection attacks. The research [37] proposed a multitask learning framework for traffic data that includes attack clustering, sample reconstruction, and sample classification. This approach extracts clustering attributes for various sorts of attacks, unique attributes for anomalous data, and latent features for classification. It also includes an enhanced Binary Cross-Entropy loss based on Focal Loss to solve imbalances in intrusion detection datasets. Experimental results showed that that method outperformed existing methods through extensive trials on the NSL-KDD, AWID, and BOT-IOT datasets. The work [38] proposed an effective intrusion detection system that employs feature interaction learning and implicit deep representation learning. The invention of a triplet-generating and learning mechanism increases representation and decision boundaries while also addressing uneven data distribution. This approach attained 3.9% accuracy, 4.1% precision, 3.9% recall, and 4.2% F1 score. The current research on intrusion detection emphasizes improving the accuracy by training and testing on the same dataset having the same network environment. The shortcoming of this kind of approach is that it achieves high accuracy during testing with similar data but cannot detect unseen attacks while testing with different datasets based on various network environments. The work in [39] evaluates the feature transferability wit,h 30, 27, 13 and 5 feature set with training and testing with AWID and AWID 3 respectively. However, the 30 and sets feature set did not achieve sustainable results its whereas the 13 and 5 feature sets achieved better results with 95 percent F1-score. Nevertheless, this work did not provide a detailed discussion on the misclassification of samples such as falsely classified instances such as false positive or false negative. This research [40] analyzed transferability in a federated setup on CIC-IDS-2017. The deep neural network is trained on one class of attack data and all available Normal data. the transferability relationships are then evaluated by testing other attack classes which were not included in training phase. Results demonstrated that proposed methodology is transferable with IDS trained with one class with the ability to attain 70% accuracy when tested with unseen attack. Another work [41] introduced a framework for evaluating the transferability of adversarial training using DNN, CNN, and LSTM on the Survival dataset. The performance of the attacks is examined by evaluating the accuracy and F1 scores of the models when tested on adversarial datasets in white-box, gray-box, and black-box scenarios. The transferability

results are satisfactory, with attacks proving effective in both gray-box and black-box scenarios. Similarly, another work [42] examined the results of transferability using Domain Adversarial Neural Network(DANN) using simulated attacks and normal operation files. The proposed methodology attained high accuracy and RMSE lower than 4.2%. This research [43] evaluates the transferability of adversarial examples. Several adversarial examples were given to well trained models and results showed that transfer attacks share similar attributes with whitebox attacks. While substantial research has emerged in the domain of attack detection, we introduce an innovative ''feature-centric'' perspective for evaluating the adaptability of deep learning models. This perspective revolves around the identification of specific features that are commonly shared and potentially versatile across a spectrum of Wi-Fi deployments, irrespective of their types or versions. In simpler terms, it is essential to figure out which specific features are useful for spotting intrusions in Wi-Fi networks and how well these features work in different Wi-Fi setups. Table 1 illustrates the previous work for this domain.

III. METHODOLOGY

The objective of this research is to explore a suitable feature set that is transferable across various network environments. Figure 1 demonstrates the methodology to evaluate the transferability across different network conditions. Algorithm 1 illustrated the methodology steps.

Algorithm 1 Intrusion Detection System (IDS) With CNN for Feature Transferability

Require: Network traffic data,Training of CNN model

1: **procedure** Preprocessing
2: Format data, Remove missing values, Normalize, Select important features
3: **end procedure**
4: **procedure** Intrusion Detection
5: Utilize 10-fold cross-validation to address imbalanced class distributions
6: Split data for training, Extract features using CNN, Classify into normal or attack (e.g., normal vs. flooding attacks)
7: **end procedure**
8: **procedure** Output and Evaluation
9: Output: Classification label, Confidence score
10: Evaluate F1-score and misclassification rate on different network environments (e.g., enterprise Wi-Fi like AWID3)
11: **end procedure**

A. DATASETS

Aegan Wi-Fi Intrusion Detection(AWID) datasets have been used for training and evaluation purposes. The primary distinction between AWID [44] (potentially referring to AWID2) and AWID3 [45] lies in their specific focuses and contexts within the realm of Wi-Fi intrusion detection. While both datasets revolve around the detection of Wi-Fi intrusions, AWID3 is purposefully tailored to cater to the specific needs of corporate applications of the Wi-Fi protocol. These corporate contexts typically demand more robust security measures. The key variances between the two datasets can be summarized as follows: **Network Complexity:** AWID3 takes into account the wide range of network architectures that are commonly encountered in commercial organizations. As a result, this dataset includes information derived from more complicated and multidimensional network configurations common in business Wi-Fi deployments. **Protocol Focus:** AWID (AWID2) focuses on scenarios related to traditional Wi-Fi intrusion detection, spanning a broad range of generic use cases. AWID3, on the other hand, is strategically directed towards the use of the Wi-Fi protocol in commercial and enterprise environments. **Enhancement of Security:** AWID3 lays a strong emphasis on adding modern security features that are routinely used in business settings. This includes the use of Protected Management Frames (PMF), a feature introduced by the 802.11w amendment that is expressly designed to improve the security of Wi-Fi networks.

While both AWID (AWID2) and AWID3 are pertinent to the domain of Wi-Fi intrusion detection, AWID3 stands out as a dataset meticulously tailored to meet the requirements of detecting intrusions in enterprise-level Wi-Fi environments. Its specialized focus on heightened security measures and the intricate network designs commonly found in the corporate sector renders it particularly relevant and valuable for real-world applications in these settings.

B. DATA PRE-PROCESSING

Before training the model, the input data must be pre-processed separately such as data formats for features including timestamps, numbers, hexadecimal digits and strings should be converted into correct format to increase the model's detection performance and convergence speed. Feature data can encompass various formats such as timestamps, numeric values, hexadecimal digits, strings, and more.

1) CLASS DISTRIBUTION

The AWID3 dataset has an imbalanced record distribution. The dataset's imbalance attribute is not changed for this study. Figure 2 and 3 depicts the distribution of the number of occurrences of normal and flooding attacks in AWID2 and AWID3. However, their imbalance property has not been modified in order to better results even with uneven distribution of classes.

2) DEALING WITH IMBALANCED DATA

Real-world scenarios often entail imbalanced wireless traffic, where the occurrence of attack frames may significantly outnumber normal frames. For instance, in 802.11 networks, DoS attacks such as de-authentication or disassociation could result in an excess of attack frames relative to regular

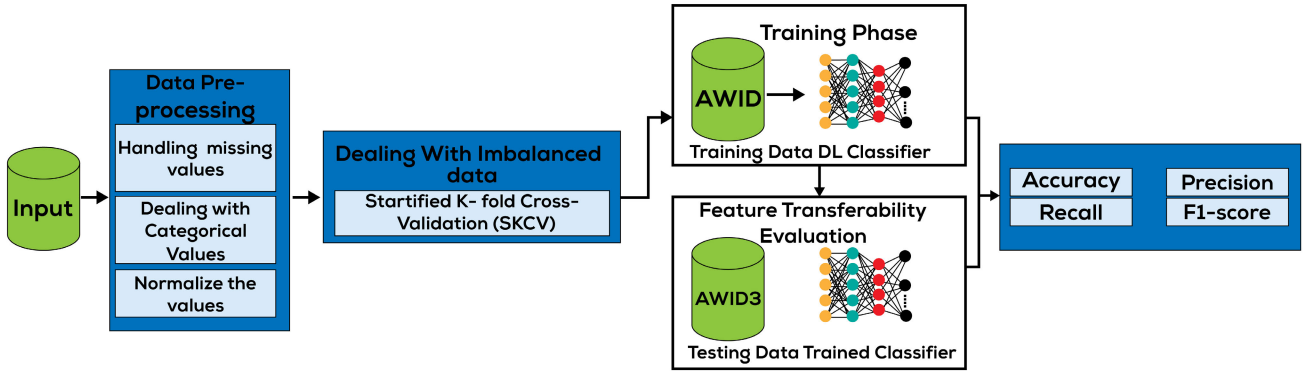

FIGURE 1. Methodology diagram.

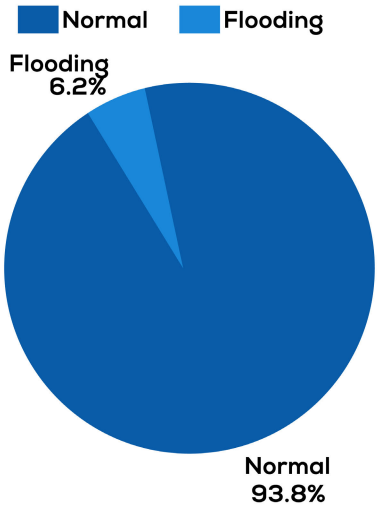

FIGURE 2. Class Distribution of AWID3.

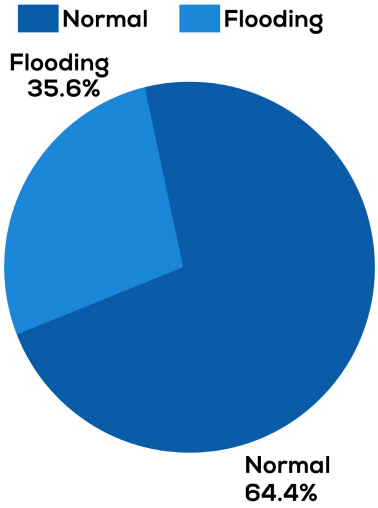

FIGURE 3. Class Distribution of AWID2.

frames. In this research,the original imbalance property of both datasets is retained, refraining from employing sampling techniques to alter their distribution. Instead, the stratified k-fold validation technique with a value of 10 is used to ensure that samples are distributed evenly throughout each class in the validation sets. It's worth mentioning that we combined the training and testing sets into a single dataset to construct the 10-fold validation sets for AWID2. Given the uneven distribution of wireless datasets, the F1-score should be prioritized and explicitly reported in work [18].

3) DEALING WITH MISSING VALUES

Missing or null values appear in AWID datasets for a variety of reasons, such as network outages or insufficient data collection during an assault. So, we remove missing and null values from our dataset for better model performance and attack detection. To address this, any occurrences with missing values are eliminated. Additionally, the attribute "wlan.fc.ds," initially consisting of hexadecimal strings, is converted to a numerical format to enhance processing and facilitate subsequent analysis and modeling procedures.

4) DATA NORMALIZATION

The ranges of feature values in the AWID datasets vary significantly. For example, the feature "radiotap.dbm_antsignal" consists of negative values such as -47,-64,-21 etc., whereas "wlan.duration" consists of high positive values such as 47,90,100. These feature range discrepancies can affect the training process. Min-max scaling is a popular approach for data normalization. This approach rationalizes a wide variety of data values by converting them to a common scale of 0 to 1 [12]. Here, in Eq. 1, $X_{\min}$ represents the minimum value of the feature X, and $X_{\max}$ denotes the maximum value for the feature.

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{1}$$

By employing min-max scaling, the model mitigates biases introduced by variables with larger scales, resulting in

enhanced accuracy and generalization. This approach promotes fair consideration of all characteristics, fostering a more robust learning process.

5) FEATURE EXTRACTION

Time based features which refers to the time series/timing of data packets such as frame.time delta (time difference between packets) and frame.time (timestamp of the packet). These features are essential for understanding the temporal behaviour of the packets to detect certain patterns such as anomlies, delays or traffic outburst. Models like Recurrent Neural Network(RNNs) or Long Short Term Memory(LSTM) are designed to capture dependencies over time and are suitable to deal with sequential data. In order to analyze these features, they need to be properly pre-processed as required for time series analysis. However in this research, these time based features are eliminated for simplification of the model's complexity which is why RNNs have not been included in the experimentation. Therefore, only those features are selected that are independent from each other which ultimately helps in minimizing multicollinearity.

For feature selection,the Random Forest algorithm was used. Gini impurity is used by Random Forest, a decision tree-based ensemble classifier, to determine the importance of the class samples within a specific node. This weight adjustment technique helps determine the significance of features in the face of unbalanced data, guaranteeing that the classifier divides the total samples of all the classes efficiently. After the classifier computed the Gini impurity $i(\tau)$ 2, a weight adjustment mechanism inside the Random Forest framework was utilized in this feature selection process to determine the feature importance concerning the unbalanced dataset output. The Gini impurity is a measure of a split's ability to divide the total number of samples of binary classes in a node.

$$i(\tau) = 1 - p_p^2 - p_n^2 \tag{2}$$

The reduction in Gini impurity resulting from each optimal split $\triangle i_f(\tau, M)$ is computed and aggregated across all M weighted trees at each node τ in the forest, where p_p represents the fraction of positive samples, p_n represents the fraction of negative samples, and N represents the total number of samples. Since each feature is analyzed independently, it is possible to thoroughly assess how well each split contributes to raising the purity of the final node designs. It is denoted in 3

$$I_g(f) = \sum_M w_n^p \left[\sum_\tau \triangle I_f(\tau, M) \right] \tag{3}$$

The Gini importance, denoted by I_g in this equation, indicates the frequency with which a particular feature (f) is selected for splitting as well as the significance of the feature's overall capacity to discriminate between classes in the binary classification job. To address unequal class distributions inside the learning method, weight w is assigned.

6) DATA SPLITTING

To counter Unbalanced Distribution of Attacks The imbalanced nature of the datasets has been retained and analyzed using 10 k-fold cross-validation.

C. CLASSIFICATION MODELS

Several machine learning models have been evaluated whereas Multilayer Perceptron (MLP) and Convolutional Neural Network (CNN) attained remarkable performance. Here is a quick introduction to give for every algorithm.

1) MULTI-LAYER PERCEPTRON (MLP)

MLP architecture is crucial in artificial neural networks. It is made up of a complex network of interconnected artificial neurons or nodes organized into layers. An input layer, one or more hidden layers, and an output layer are included in these layers.

Input Layer: This is the first layer that allows raw data or features into the neural network. This layer's nodes each relate to a different input characteristic or variable.

Hidden Layers: The MLP's hidden layers are located between the input and output layers.

Output Layer: The final layer, known as the output layer, is responsible for producing the network's predictions or classifications.

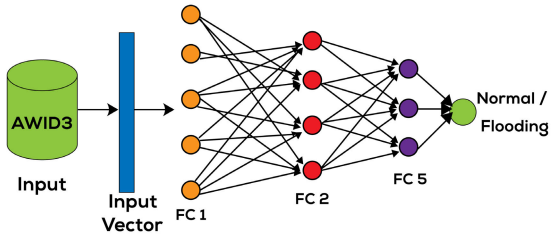

FIGURE 4. Illustration of multi-layer perceptron.

The input data for using an MLP model for intrusion detection in the AWID dataset includes critical information such as packet sizes, flow durations, source and destination IP addresses, port numbers, protocols, and numerous network-related variables. The MLP model goes through a training procedure using labeled data to distinguish between regular network operations and instances of intrusion in the dataset as shown in Figure 4. While ''FC'' stands for ''Fully Connected'' layer. Specifically, FC1 refers to the first fully connected layer, FC2 to the second fully connected layer, and FC3 to the third fully connected layer.

a: TRAINING OF MLP CLASSIFICATION

In order to perform classification, the MLP model is trained to label specific occurrences of network traffic as either ''normal'' or ''flooding'', and set values 0 for normal class and 1 for flooding class. Each labeled instance in the training set is a sequence of network traffic data collected during a given time range. frame.len, radiotap.length, radiotap.dbm

and signal are included in the features of the traffic data. To implement an MLP (Multi-Layer Perceptron) model for intrusion detection within the AWID dataset, key MAC address features are given the input data. Subsequently, using labeled examples, the MLP model is trained using the Adam optimizer and the Rectified Linear Unit (ReLU) activation function, which is distributed across five layers. The goal of this training procedure is to teach the model whether a given sequence represents normal network activity or indicates the existence of an intrusion, specifically flooding attacks. This architecture is made up of numerous dense (completely linked) layers with ReLU activation functions interspersed by regularization dropout layers. After the initial thick layer, a dropout layer with a dropout rate of 20 percent is applied to prevent overfitting. Dropout improves model generalization by randomly deactivating a subset of neurons during training. Following that, the data is routed through another dense layer with 128 neurons and ReLU activation, followed by a dropout layer with the same dropout rate. This pattern is repeated with denser layers, including one with 64 neurons, one with 32 neurons, and one with 16 neurons, all of which use ReLU activation functions. Throughout the training phase, the MLP model learns to recognize detailed patterns within input sequences, allowing for a clear distinction between normal network behaviors and the manifestation of flooding attacks. By leveraging previous events inside network traffic data, the model improves its ability to capture the different features and behaviors quirks associated with flooding attacks, thus enhancing its intrusion detection skills within the AWID dataset.

2) CONVOLUTIONAL NEURAL NETWORK (CNN)

CNN is designed primarily to detect complicated spatial and temporal patterns in datasets. When applied to intrusion detection, CNNs are essential for grasping complicated patterns within the dataset's different attributes. The critical convolutional layer, which performs a foundational role, is located at the core of CNN design. Following the feature extraction step, pooling layers are used to further scrutinize the data, with the twin goal of improving data analysis and lowering the spatial dimensions of the resulting feature maps. Then, fully connected layers take over, resulting in the ultimate classification. It's important to note that the weights of the filters, as well as those of the fully connected layers, are polished and fine-tuned by an arduous training procedure, which frequently employs a labeled dataset.

This model is depicted graphically in Figure 5 below:

It is important to note that CNN does not detect these patterns indiscriminately; rather, it learns them through rigorous training on labeled datasets. For example, in the context of intrusion detection, if the dataset includes network packets as independent features, CNN can deduce patterns from the header fields of these packets. These fields could include crucial information including source and destination IP addresses, port numbers, and protocol kinds. By adeptly absorbing and recognizing these patterns, the CNN transforms into a skilled guardian, capable of identifying anomalies within network traffic that may indicate future intrusions. It has the following components:

a: INPUT LAYER

In the case of intrusion classification, the input layer of a CNN receives data from network traffic or log files. This data could involve packet sizes, durations, source and destination IP addresses, port numbers, and protocol types.

Convolutional Layer: In intrusion classification, the convolutional layer applies filters or kernels to input data to identify patterns that indicate various types of network attacks. These patterns could indicate anomalous traffic behavior, unreliable connections, or known attack features.

b: ACTIVATION LAYER (RELU)

Following convolution, the ReLU activation function is implemented to the feature maps resulting from the convolutional layer. This provides nonlinearity to the model, allowing it to capture complicated correlations between input features and intrusion classes.

c: POOLING LAYER

The pooling layer in intrusion classification decreases the spatial dimensionality of feature maps while keeping critical information about observed patterns. This aids in focusing on the most important elements for intrusion detection while eliminating extraneous details.

Output Layer: The output layer is designed to take the features extracted and processed by the preceding layers of the neural network and produce probabilities for the two possible classes in the AWID dataset. These probabilities are generated using the sigmoid activation function, allowing the model to make classification decisions.

d: TRAINING OF CNN CLASSIFICATION

The first layer of this model is a 1D convolution (Conv1D) with 128 filters, a kernel size of 3, and ReLU activation. It uses ''same'' padding to maintain the input sequence length of (8, 1). This suggests the model is designed for sequences with 8 elements. Following the convolution is a max-pooling layer (MaxPooling1D) that reduces the sequence length by half (pool size of 2) while capturing the most significant features. A second convolutional layer with 64 filters, kernel size 3, ''same'' padding, and ReLU activation is applied. Similar to the first layer, it maintains the sequence length. Another max-pooling layer follows this convolution. A flatten layer then transforms the 2D feature maps from the convolutional layers into a 1D vector, preparing the data for fully connected layers. These dense layers process the flattened vector to make the final prediction. These layers of data learn to recognize patterns and relationships. The ReLU activation function is used in the first dense layer, which includes 128 neurons. Because it assists the network in modeling complicated patterns, ReLU is a popular choice for activation functions in hidden layers. The second dense layer contains

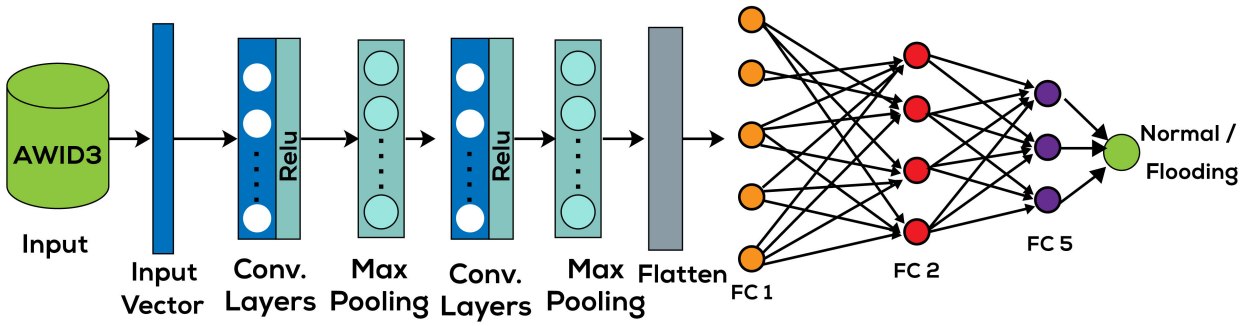

FIGURE 5. Illustration of a convolutional neural network.

32 neurons that are activated by ReLU. The third dense layer contains 16 neurons that are activated by ReLU. Finally, an output layer with two neurons and a softmax activation function is present. Softmax is used to determine class probabilities in this classification (normal or flooding attack).

D. FEATURE TRANSFERABILITY

In the realm of IDS, the assessment of their effectiveness relies on key metrics like precision, recall, F1 score, and AUC score. As expected, when these IDS models are individually trained and tested on well-known datasets like AWID and AWID3, they exhibit remarkable performance, boasting high F1 scores ranging from 0.98 to 1 and AUC scores ranging from 0.97 to 0.99. However, the true challenge arises when attempting to transfer these high-performing models to entirely new and unseen datasets. Here, the performance landscape becomes more diverse, suggesting that the outstanding performance achieved on a particular dataset doesn't automatically guarantee the model's ability to generalize to novel datasets within unique network environments. This brings us to a pivotal question: Can the set of selected features seamlessly adapt across different datasets? To investigate this, the model, having been trained on these chosen features, undergoes rigorous testing using previously unseen network traffic data. This evaluation extends to real-time scenarios and diverse network environments, providing researchers with insights into the enduring effectiveness and wide applicability of these selected features beyond the boundaries of the original dataset. This evaluation is of paramount importance as it validates the feasibility and adaptability of the proposed cyber-attack detection model in dynamic and varied network conditions. In essence, it tests whether the model can maintain its effectiveness and relevance when faced with the complexities of real-world network scenarios. In this research, feature transferability is evaluated by training deep learning models on selected features of AWID dataset and testing with AWID3. AWID3 is the reformed version of AWID dataset and designed on enterprise network environment. Consequently, the purpose of testing feature transferability is to discover the strongest and most advantageous features that can be efficiently transferred or generalized across various network environments, especially between a general network environment and an enterprise network environment. Particularly, this includes evaluation of the models's ability to maintain its efficacy when applied to both a general network environment(AWID) and an enterprise network environment(AWID3). In this way, this study attempts to discover features that remain successful when transferred from a general WiFi network environment to a corporate WiFi network environment in the context of AWID3, which focuses on enterprise versions of the protocol. The inclusion of stronger security measures and various network topologies in the workplace context could require the use of particular features to detect intrusions efficiently. Transferability is essential for developing Wi-Fi IDS models that can accurately detect intrusion patterns across a wide range of network scenarios, from small residential networks to large industrial and enterprise networks. Using data from AWID2, we trained a couple of classifiers, and we evaluated their performance using AWID3. For training, we employed the AWID2-CLS-R data, which contained just the Normal and Flooding classes. the AWID-CLS-F-Trn and AWID-CLS-F-Tst files, which together comprised the whole AWID2 dataset, were merged. De-authentication attacks are the common attack in both datasets. The training set consisted of just the Normal and Flooding classes, Next, we tested the classifiers on the Normal and De-authentication traffic found in the AWID3 De-authentication attack instances. In this research, de-authentication attacks are denoted by flooding.od

IV. RESULTS AND DISCUSSION

The application of Multilayer Perceptron (MLP) and Convolutional Neural Network (CNN) involves the utilization of eight MAC address features that are independent of the channel used for capturing network traffic.

A. PARAMETER CONFIGURATION

The parameter values for CNN methods can vary based on implementation and issue domains. These settings are

not exhaustive and may require additional adjustment and testing depending on the specific situation and dataset. When choosing parameter values for CNNs and MLPs, it is crucial to carefully assess the trade-offs between exploration and exploitation, consider network design, and account for other relevant factors.

1) LEARNING RATE

In the MLP model, the optimizer employed was the Adaptive Moment Estimation (Adam) method, with a learning rate set at 0.0001. For both the CNN and MLP models, a learning rate of 0.001 was utilized with the Adam optimizer.

2) OVERFITTING PREVENTION

Various techniques were implemented to counteract overfitting. Initially, an early stopping strategy was employed, permitting the models to undergo two additional rounds before cessation to mitigate overfitting to the training data. Additionally, dropout layers were incorporated, randomly deactivating specific neurons during training to prevent their dominance in the learning process.

3) ACTIVATION FUNCTION

The Rectified Linear Unit (ReLU), a simple and effective activation function, is used in this research. ReLU displays monotonic behavior, which means that both the function and its derivative follow a consistent pattern. When given negative input values, ReLU returns 0, successfully suppressing negative signals. In contrast, for positive input values (x), ReLU just outputs the input value without any changes. As a result, ReLU's output ranges from 0 to infinity, providing a large range of possible values. Eq. 4 expresses the mathematical transition made by the ReLU activation function. This transformation accelerates model generalization and improves overall accuracy by allowing for rapid learning of complicated patterns in the data.

$$f(x) = \max \begin{cases} 0 & \text{for all } x < 0 \\ x & \text{for all } x > 0 \end{cases} \tag{4}$$

4) BATCH SIZE

The quantity of was selected randomly from the replay buffer during each training iteration, with typical values ranging from 32 to 256.

5) NUMBER OF HIDDEN LAYERS

The count of hidden layers in the neural network architecture between input and output layers. This varies depending on the complexity of the task, typically ranging from one to three layers.

6) NUMBER OF FILTERS

The number of filters specifies the number of channels in each CNN layer. In this research, 2 convolutional layers have been use. 128 and 64 number of filters are used in in these layers correspondingly. This means that CNN will learn 128 features in first convolutional layer from input and 64 features in second convolutional layer from output of the previous layer.

7) KERNEL SIZE

It is the length of 1D convolutional window. 3×3 kernel size is used in this research to capture local patterns and extract relevant features.

8) PADDING

It specifies how the borders of the whole data are handles. In this research, padding is set to "same" which means input and output has same length by adding zeros at the edges.

9) MAX POOLING

It selects the highest value from each pool of values leading to dimensionality reduction in feature maps. A pool size of 2 has been selected in this work, which means 2 values will be examined and the maximum will be selected.

B. FEATURE SELECTION

Table 2 illustrates the MAC and Physical layer features extracted using Random Forest. The MAC attributes chosen were carefully chosen for their relevance and usefulness in aiding the creation of an effective Wi-Fi cyber threat detection system.

1) FRAME LENGTH

It aids in the detection of abnormal packet sizes which is an essential feature in detection of flooding attacks. These attacks involve ample amount of frames to keep the system occupied. So if the Wi-Fi network.

2) RADIOTAP.DBM.ANTSIGNAL

This feature measures the strength of the Wi-Fi signals at the receiving end. Attackers who are distant from the main location have weaker signals. If frequently weak signals are noticed in the setup then it seems to have been attacked.

3) WLAN.DURATION

If the frame transmission is taking an unusually long time then it could jam the network and can be the indication of flooding attack in the network. radiotap.channel.freq: the frequent variation in channel frequency can indicate an anomaly often associated with attacks.

4) WLAN.FC.TYPE/SUBTYPE

These are flags. These flags identify the type of frames such as deauthentication and disassociation frames etc. The flooding attack in this dataset also includes network traffic of deauthentication and disassociation attacks. Now assume that a large number of deauthentication or disassociation frames are transmitted over the network, this pattern can indicate an attack as attackers use these frames to disconnection and reconnection to rogue access points.

5) WLAN.FC.RETRY

This flag can indicate the deviation in a usual pattern when the frames are retransmitted. For example, a sudden rise in retry frames can indicate that the devices are encountering any deviation.

6) WLAN.FC.MOREDATA

if a **then** a device shows more data flag repetitively at a time then it can be an indication of attack.

These features are preferred over other features as they capture a wide range of characteristics of the network traffic and analyze high-level traffic patterns to detect potential attacks. The features like radiotap.present.tstf, radiotap.length or other specialized flag-like features such as radiotap.pwrmgt provide limited information. These features provide highly specific information of the network behavior and can be tightly related to certain cases only. The extracted features are suitable for both home and enterprise-based environments. For example, radiotap.dbm.antsignal is crucial for reliable connection whether it is a single access point for home-based environment or multiple access points in an enterprise environment.frame.len is essential in both environments for monitoring the bandwidth. wlan.fc.type identifies the type of traffic such as management or control frames for both network environments. wlan.fc.subtype determines the purpose of frame such as beacon frame, authentication or association requests etc. This feature aid in tracking associations and disassociations caused by attackers in home based environment whereas it aids in recognizing association requests between access points in an enterprise environment [44].

TABLE 2. MAC and physical layer features.

Feature Name	Description
Frame.len	Frame Length
Radiotap.dbm_antsignal	Present flag antenna signal (dbm)
Wlan.duration	Duration time
Radiotap.channel.freq	Channel frequency value
Wlan.fc.type	Type- Flag
Wlan.fc.subtype	Subtype-Flag
Wlan.fc.retry	Retry-flag
Wlan.fc.moredata	More data-flag

C. PERFORMANCE EVALUATION METRICS

The evaluation of the CNN and MLP-based model's effectiveness and precision in identifying and categorizing network intrusions is conducted as part of the performance assessment for feature transferability in Wi-Fi intrusion detection.

1) CONFUSION MATRICES

The assessment of the proposed approach for cyberattack detection involves the utilization of a confusion matrix, providing crucial insights into the system's performance. The four fundamental metrics in the confusion matrix include true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

a: TRUE POSITIVES

True positives (TP) represent instances where the system correctly identified incidents as cyberattacks. These are situations where the system has successfully recognized the presence of a cyberattack.

b: TRUE NEGATIVES

True negatives (TN) are the instances correctly categorized as normal or routine network traffic. These occurrences are accurately identified as benign, showcasing the system's ability to distinguish between regular and malicious data.

c: FALSE POSITIVES

False positives (FP) denote the instances that are inaccurately categorized as a form of cyberattack. These incidents are wrongly identified as assaults when they are not, leading to misclassification.

d: FALSE NEGATIVE

False negatives (FN) represent the instances of cyberattacks that escape detection by the intrusion detection system. As these occurrences are not flagged as attacks, detrimental actions remain unidentified.

An evaluation of the proposed methodology's effectiveness involves examining its ability to accurately identify cyberattacks (TP), appropriately classify normal traffic (TN), minimize false alarms (FP), and prevent overlooking legitimate attacks (FN). Analyzing these metrics provides a comprehensive assessment of the system's performance, aiding in the determination of its reliability and efficiency in detecting cyberattacks.

2) EVALUATION MEASURES

This study employed various evaluation criteria to assess the system's performance, which depended on the characteristics of the confusion matrix. In this research, the concept of transferability is evaluated by reusing learned features across diverse validated dataset with in similar domain which is why standard performance metrics sufficiently capture the performance of feature transfer instead of transfer accuracy or domain adaptation methods. These metrics encompass precision, recall, and the F1 measure.

a: ACCURACY

A commonly employed metric for gauging the overall accuracy of the system's predictions is to calculate the ratio of correctly classified instances (TP and TN) to the total number of instances using eq. 5. This metric offers insight into the system's performance across both positive and negative classifications.

$$Accuracy = \frac{True_p + True_N}{True_p + True_N + False_p + False_N} \quad (5)$$

Transferability evaluation measures the accuracy of categorizing cases, including both normal and cyberattacks, in the dataset. It provides an assessment of how well the CNN and MLP-based models make correct predictions overall.

b: PRECISION

Cyberattack detection precision is the ratio of correctly identified cyberattacks to all instances categorized as intrusions by the CNN and MLP-based model.It is calculated using eq. 6 This metric evaluates the model's capability to accurately discern intrusions, ensuring that it does not misclassify regular cases as cyberattacks.

$$Precision = \frac{True_P}{True_P + False_P} \tag{6}$$

c: RECALL

The CNN and MLP-based model's recall, also known as sensitivity or true positive rate, quantifies the percentage of accurately identified cyberattacks using eq. 7 This metric assesses the model's efficacy in detecting intrusions among all the actual incursions present in the dataset.

$$Recall = \frac{True_P}{True_P + False_N} \tag{7}$$

d: F1 SCORE

The F1 score integrates accuracy and recall into a unified metric, providing a more equitable assessment of the system's performance. It represents the harmonic mean of accuracy and recall, assigning equal importance to both measures. In situations where there is an uneven distribution between positive and negative instances, the F1 measure becomes particularly valuable. This score serves as a comprehensive performance metric, encapsulating both accuracy and recall, to gauge the overall success of the CNN-based model in detecting cyberattacks and calculated by eq. 8.

$$F1score = \frac{2True_P}{2True_P + False_P + False_N} \tag{8}$$

D. PERFORMANCE ANALYSIS

1) TRAINING PHASE ANALYSIS

CNN-based network intrusion detection techniques outperform traditional approaches in terms of f1-score and classification accuracy. The benefit arises from their capacity to efficiently extract local feature information [13]. This advantage enables CNNs to outperform MLPs in classification tasks. During training with the AWID dataset, CNN exhibited an f1-score of 99.96 and an accuracy of 99.94%, slightly surpassing the MLP, which achieved 99.83% and 99.84% respectively. Precision and recall also follow this trend, with CNN achieving 99.92% and 99.95% and MLP reaching 99.81% and 99.80% for precision and recall respectively. The AUC-score is 99.94% for CNN and 99.80% for MLP.

Table 3 presents the training analysis of the two deep learning models, MLP and CNN, conducted through 10 k-fold cross-validation on the AWID dataset. The results demonstrate that the selected features exhibit robustness and efficiency in detecting attacks, achieving a 99% f1-score and accuracy rate with the AWID dataset.

TABLE 3. Training evaluation on AWID dataset.

Performance Metrics	CNN	MLP
Accuracy	99.94	99.84
Precision	99.92	99.81
Recall	99.95	99.80
F1-score	99.96	99.83
AUC score	99.94	99.80

Figures 6, 7 show the accuracy and loss graphs of CNN and Figures 8, 9 illustrate the MLP respectively. Figures 6, 7 show that training and validation loss and accuracy of CNN are equivalent whereas train loss and accuracy of MLP are lower than validation loss and accuracy. This implies that the MLP model is leveraged to overfitting whereas CNN proved to be the ideal model for transferability evaluation.

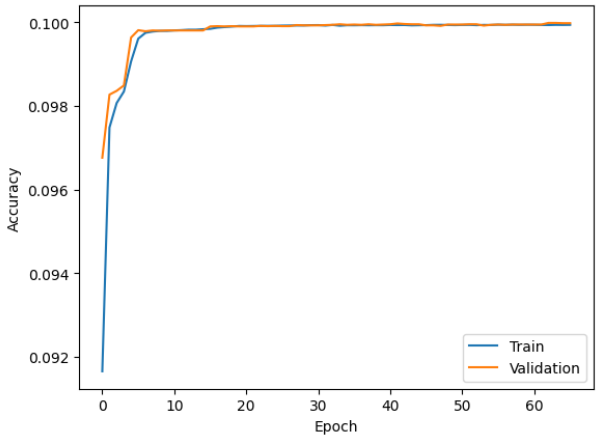

FIGURE 6. CNN Accuracy Graph.

2) TESTING WITH AWID3 TO EVALUATE FEATURE TRANSFERABILITY

Our primary objective is to evaluate the transferability of the selected attributes across diverse network settings. Despite this focus, we diligently conducted and reported evaluations that achieved exceptionally high predicted accuracy. The examination of feature transferability is crucial as it enables us to discern whether the chosen features can be effectively utilized in network contexts beyond their original creation environment. This research contributes valuable insights into the generalizability and robustness of the feature set, allowing us to gauge the potential application of the intrusion detection system across various scenarios. Table 4 illustrates the analysis of feature transferability in terms of accuracy, precision, recall, and F1-score. CNN exhibited promising

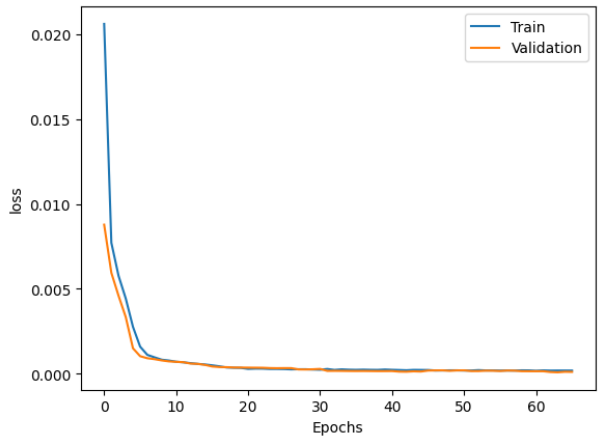

FIGURE 7. CNN loss graph.

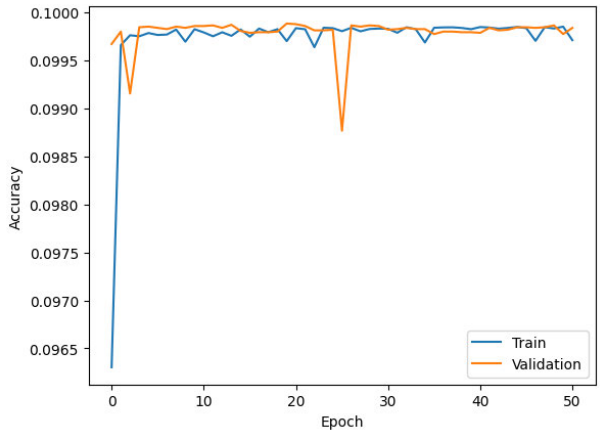

FIGURE 8. MLP accuracy graph.

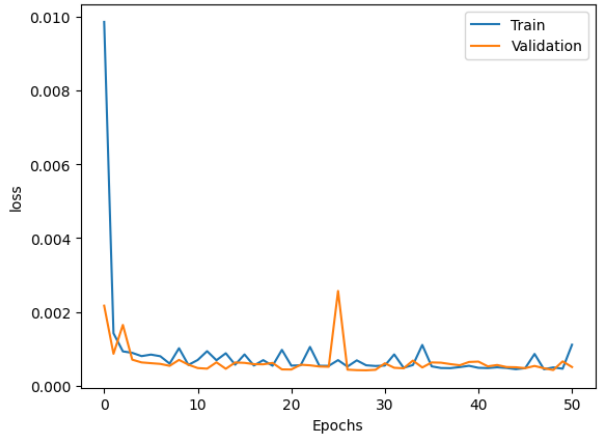

FIGURE 9. MLP loss graph.

results, achieving an impressive accuracy and F1-score of 97.28% and 98.53%.

When evaluating the performance of the models on unseen data from different network environments, CNN performed

TABLE 4. Testing evaluation on AWID3 dataset.

Models	Accuracy	Precision	Recall	F1-score
CNN	97.28%	97.71%	99.37%	98.53%
MLP	96.52%	75.80%	94.06%	83.94%

better than MLP when comparing their respective performances on unseen data from various network configurations demonstrated in Figures 10 and 11. CNN specifically showed a notable decrease in incorrect classifications, especially when correctly differentiating flooding attacks from regular traffic. The difference in performance between CNN and MLP was almost 4000 cases, highlighting CNN's improved accuracy and flexibility in various network scenarios. MLP performs well in classifying normal traffic with high true negative count. However, as the dataset is imbalanced, 18,180 instances are falsely classified as normal instances using MLP leading to a high rate of false negatives for the minority class. Due to class imbalance, false positives can also be affected as normal traffic is falsely classified as an attack overwhelming the IDS. Here, the number of false positive instances is 3593 which is relatively small compared to true negatives. CNN has shown better results in terms of class imbalance as it has a comparatively lower number of false negatives. Furthermore, it has higher true negatives which means CNN attained better accuracy for normal traffic instances. This way CNN is performing better under class imbalance problem as it correctly classifies minority class without increasing false positives.

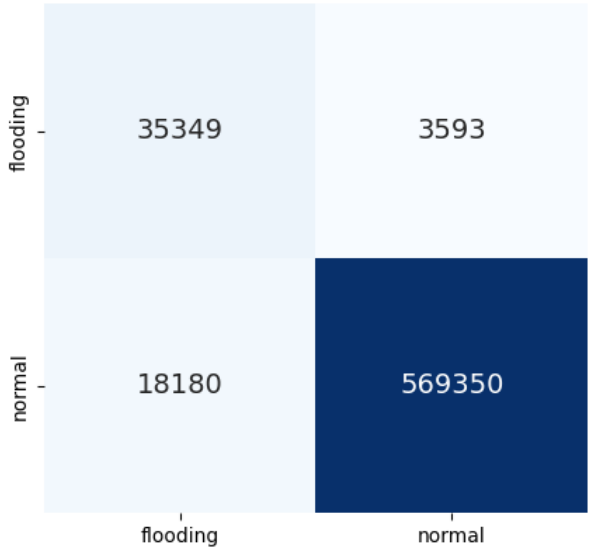

FIGURE 10. MLP with testing data.

The promising performance of both models on the original dataset indicates their potential to be effectively applied in real-world intrusion detection scenarios. Nevertheless, CNN handles spatial patterns using convolutional layers that can capture local dependencies in the network traffic by

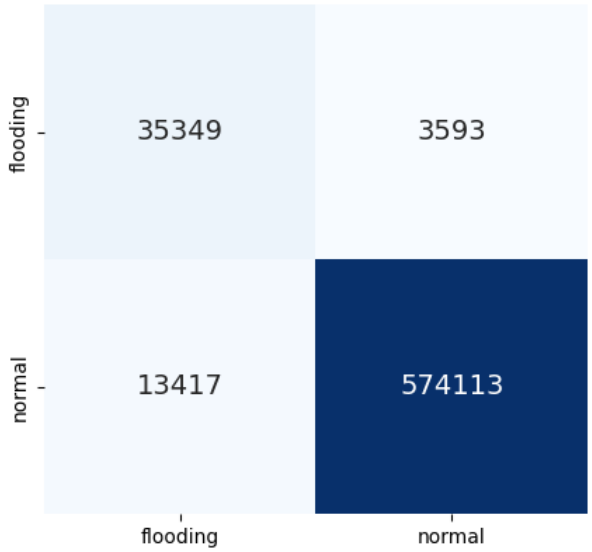

FIGURE 11. CNN with testing data.

TABLE 5. Performance comparison with state of the art techniques.

Reference	Technique	Dataset	Result	Transferability
[14]	EBDM-DNN	AWID	F1-score = 99.25%	No
[16]	A.E-SAC	AWID	F1-score = 98.9%	No
[19]	DT, ET	AWID AWID3	F1-score = 95.93%	Yes
Proposed Method	CNN	AWID AWID3	Acc = 97.28% F1-score = 98.53%	Yes

applying filters. These filters can detect any deviation in the network traffic. Unlike MLP which handles all the features independently, CNN look for local dependencies [46]. The superior transferability of CNN, as demonstrated by its capacity to handle unseen data from different network environments more accurately, suggests that it may be a more suitable choice for practical applications in dynamic and varied network settings. In conclusion, both CNN and MLP exhibit strong predictive capabilities, but the higher transferability and reduced misclassification of attacks observed in CNN underscore its prominence as a reliable choice for intrusion detection across diverse network conditions.

E. COMPARISON WITH STATE OF ART TECHNIQUES

Current research on Wi-Fi IDS lacks the exploration and evaluation of the transferability of features for different Wi-Fi network environments. Table 5 compares the proposed methodology with state-of-the-art techniques. The previous research with AWID3 is based on supervised learning and cannot comprehend the ever-changing nature of cyber-attacks.

V. CONCLUSION

Our study introduced two models, CNN and MLP, they achieved highly promising results in cyber-attack detection, with F1-scores and accuracies reaching up to 97%. This signifies the efficacy of both models in accurately recognizing and categorizing attacks within the dataset they were trained on. While the commendable performance of all models on the original dataset suggests their potential for effective application in real-world intrusion detection scenarios, the superior transferability of CNN, as evidenced by its more accurate handling of unseen data from different network environments, suggests it may be a more suitable choice for practical applications in dynamic and varied network settings. In conclusion, both CNN and MLP demonstrate excellent prediction skills; nevertheless, CNN is more dependable for intrusion detection under a variety of network situations due to its higher transferability with higher f1-score and decreased misclassification of attacks. Subsequent investigations inside this field ought to go into methods for boosting model transferability, guaranteeing effectiveness in practical settings with diverse network attributes. Future research in this domain should further explore techniques to enhance the transferability of models to ensure their efficacy in real-world environments with varying network characteristics.

Appendixes, if needed, appear before the acknowledgment.

ACKNOWLEDGMENT

The authors acknowledge with thanks the University of Jeddah for its technical support.

REFERENCES

[1] M. A. Jabbar, R. Aluvalu, and S. S. Reddy, "RFAODE: A novel ensemble intrusion detection system," *Proc. Comput. Sci.*, vol. 115, pp. 226–234, Jan. 2017, doi: 10.1016/j.procs.2017.09.129.

[2] Gmcdouga. *Check Point Research Reports a 38 Percent Increase in 2022 Global Cyberattacks*. Check Point Blog. Accessed: May 22, 2023. [Online]. Available: https://blog.checkpoint.com/2023/01/05/38-increase-in-2022-global-cyberattacks/

[3] AAG IT Services. (May 2023). *The Latest Cyber Crime Statistics*. Accessed: May 22, 2023. [Online]. Available: https://aag-it.com/the-latest-cyber-crime-statistics/

[4] N. Fluschnik, F. Ojeda, T. Zeller, T. Jørgensen, K. Kuulasmaa, P. M. Becher, C. Sinning, S. Blankenberg, and D. Westermann, "Predictive value of long-term changes of growth differentiation factor-15 over a 27-year-period for heart failure and death due to coronary heart disease," *PLoS ONE*, vol. 13, no. 5, May 2018, Art. no. e0197497.

[5] M. Masdari and H. Khezri, "A survey and taxonomy of the fuzzy signature-based intrusion detection systems," *Appl. Soft Comput.*, vol. 92, Jul. 2020, Art. no. 106301, doi: 10.1016/j.asoc.2020.106301.

[6] P. Dini and S. Saponara, "Analysis, design, and comparison of machine-learning techniques for networking intrusion detection," *Designs*, vol. 5, no. 1, p. 9, Feb. 2021, doi: 10.3390/designs5010009.

[7] G. S. C. Kumar, R. K. Kumar, K. P. V. Kumar, N. R. Sai, and M. Brahmaiah, "Deep residual convolutional neural network: An efficient technique for intrusion detection system," *Expert Syst. Appl.*, vol. 238, Mar. 2024, Art. no. 121912, doi: 10.1016/j.eswa.2023.121912.

[8] M. S. Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, "Detecting abnormal traffic in large-scale networks," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Oct. 2020, pp. 1–7, doi: 10.1109/ISNCC49221.2020.9297358.

[9] L. Ashiku and C. Dagli, "Network intrusion detection system using deep learning," *Proc. Comput. Sci.*, vol. 185, pp. 239–247, Jan. 2021, doi: 10.1016/j.procs.2021.05.025.

[10] F. Laghrissi, S. Douzi, K. Douzi, and B. Hssina, "Intrusion detection systems using long short-term memory (LSTM)," *J. Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00448-4.

[11] Y. N. Kunang, S. Nurmaini, D. Stiawan, and B. Y. Suprapto, "Attack classification of an intrusion detection system using deep learning and hyperparameter optimization," *J. Inf. Secur. Appl.*, vol. 58, May 2021, Art. no. 102804, doi: 10.1016/j.jisa.2021.102804.

[12] A. V. Turukmane and R. Devendiran, "M-MultiSVM: An efficient feature selection assisted network intrusion detection system using machine learning," *Comput. Secur.*, vol. 137, Feb. 2024, Art. no. 103587, doi: 10.1016/j.cose.2023.103587.

[13] N. O. Aljehane, H. A. Mengash, S. B. H. Hassine, F. A. Alotaibi, A. S. Salama, and S. Abdelbagi, "Optimizing intrusion detection using intelligent feature selection with machine learning model," *Alexandria Eng. J.*, vol. 91, pp. 39–49, Mar. 2024, doi: 10.1016/j.aej.2024.01.073.

[14] A. Agrawal, U. Chatterjee, and R. R. Maiti, "KTRACKER: Passively tracking Krack using ML model," in *Proc. 12th ACM Conf. Data Appl. Secur. Privacy*, Apr. 2022, pp. 364–366, doi: 10.1145/3508398.3519360.

[15] C. H. Tseng and Y.-T. Chang, "EBDM: Ensemble binary detection models for multi-class wireless intrusion detection based on deep neural network," *Comput. Secur.*, vol. 133, Oct. 2023, Art. no. 103419, doi: 10.1016/j.cose.2023.103419.

[16] S. M. Kasongo and Y. Sun, "A deep learning method with wrapper based feature extraction for wireless intrusion detection system," *Comput. Secur.*, vol. 92, May 2020, Art. no. 101752, doi: 10.1016/j.cose.2020.101752.

[17] Z. Li, C. Huang, S. Deng, W. Qiu, and X. Gao, "A soft actor-critic reinforcement learning algorithm for network intrusion detection," *Comput. Secur.*, vol. 135, Dec. 2023, Art. no. 103502, doi: 10.1016/j.cose.2023.103502.

[18] R. B. Basnet, R. Shash, C. Johnson, L. Walgren, and T. Doleck, "Towards detecting and classifying network intrusion traffic using deep learning frameworks," *J. Internet Services Inf. Secur.*, vol. 9, no. 4, pp. 1–17, 2019.

[19] N. Yadav, S. Pande, A. Khamparia, and D. Gupta, "Intrusion detection system on IoT with 5G network using deep learning," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–13, Mar. 2022, doi: 10.1155/2022/9304689.

[20] H. Sadia, S. Farhan, Y. U. Haq, R. Sana, T. Mahmood, S. A. O. Bahaj, and A. R. Khan, "Intrusion detection system for wireless sensor networks: A machine learning based approach," *IEEE Access*, vol. 12, pp. 52565–52582, 2024, doi: 10.1109/ACCESS.2024.3380014.

[21] Z. Salah and E. A. Elsoud, Jul. 2023, "Enhancing intrusion detection in 5G and IoT environments: A comprehensive machine learning approach leveraging AWID3 dataset," doi: 10.20944/preprints202307.1565.v1.

[22] A. A. Bhutta, M. U. Nisa, and A. N. Mian, "Lightweight real-time WiFi-based intrusion detection system using LightGBM," *Wireless Netw.*, vol. 30, no. 2, pp. 749–761, Oct. 2023, doi: 10.1007/s11276-023-03516-0.

[23] L. M. da Silva, V. M. Andreghetti, R. A. F. Romero, and K. R. L. J. C. Branco, "Analysis and identification of evil twin attack through data science techniques using AWID3 dataset," in *Proc. 6th Int. Conf. Mach. Learn. Mach. Intell.*, Oct. 2023, pp. 128–135, doi: 10.1145/3635638.3635665.

[24] L. Narengbam and S. Dey, "WiFi intrusion detection using artificial neurons with bio-inspired optimization algorithm," *Proc. Comput. Sci.*, vol. 218, pp. 1238–1246, Jan. 2023, doi: 10.1016/j.procs.2023.01.102.

[25] H. Tan, L. Wang, D. Zhu, and J. Deng, "Intrusion detection based on adaptive sample distribution dual-experience replay reinforcement learning," *Mathematics*, vol. 12, no. 7, p. 948, Mar. 2024, doi: 10.3390/math12070948.

[26] S. M. Kasongo and Y. Sun, "A deep gated recurrent unit based model for wireless intrusion detection system," *ICT Exp.*, vol. 7, no. 1, pp. 81–87, Mar. 2021, doi: 10.1016/j.icte.2020.03.002.

[27] K. Sethi, E. S. Rupesh, R. Kumar, P. Bera, and Y. V. Madhav, "A context-aware robust intrusion detection system: A reinforcement learning-based approach," *Int. J. Inf. Secur.*, vol. 19, no. 6, pp. 657–678, Dec. 2019, doi: 10.1007/s10207-019-00482-7.

[28] P. R. Kannari, N. C. Shariff, and R. L. Biradar, "Network intrusion detection using sparse autoencoder with swish-PReLU activation model," *J. Ambient Intell. Humanized Comput.*, Mar. 2021, doi: 10.1007/s12652-021-03077-0.

[29] Q. Duan, X. Wei, J. Fan, L. Yu, and Y. Hu, "CNN-based intrusion classification for IEEE 802.11 wireless networks," in *Proc. IEEE 6th Int. Conf. Comput. Commun. (ICCC)*, Chengdu, China, Dec. 2020, pp. 830–833, doi: 10.1109/ICCC51575.2020.9345293.

[30] V. V. Thang and F. F. Pashchenko, "Multistage system-based machine learning techniques for intrusion detection in WiFi network," *J. Comput. Netw. Commun.*, vol. 2019, pp. 1–13, Apr. 2019, doi: 10.1155/2019/4708201.

[31] M. E. Aminanto, H. C. Tanuwidjaja, P. D. Yoo, and K. Kim, "Wi-Fi intrusion detection using weighted-feature selection for neural networks classifier," in *Proc. Int. Workshop Big Data Inf. Secur. (IWBIS)*, Sep. 2017, pp. 99–104, doi: 10.1109/IWBIS.2017.8275109.

[32] Y. Qin, B. Li, M. Yang, and Z. Yan, "Attack detection for wireless enterprise network: A machine learning approach," in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC)*, Qingdao, China, Sep. 2018, pp. 1–6, doi: 10.1109/ICSPCC.2018.8567797.

[33] R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. Alessa, "Effective features selection and machine learning classifiers for improved wireless intrusion detection," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Rome, Italy, Jun. 2018, pp. 1–6, doi: 10.1109/ISNCC.2018.8530969.

[34] S. Lei, C. Xia, Z. Li, X. Li, and T. Wang, "HNN: A novel model to study the intrusion detection based on multi-feature correlation and temporal–spatial analysis," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 4, pp. 3257–3274, Oct. 2021, doi: 10.1109/TNSE.2021.3109644.

[35] S. Rezvy, Y. Luo, M. Petridis, A. Lasebae, and T. Zebin, "An efficient deep learning model for intrusion classification and prediction in 5G and IoT networks," in *Proc. 53rd Annu. Conf. Inf. Sci. Syst. (CISS)*, Baltimore, MD, USA, Mar. 2019, pp. 1–6, doi: 10.1109/CISS.2019.8693059.

[36] T. Gaber, A. El-Ghamry, and A. E. Hassanien, "Injection attack detection using machine learning for smart IoT applications," *Phys. Commun.*, vol. 52, Jun. 2022, Art. no. 101685, doi: 10.1016/j.phycom.2022.101685.

[37] M. Asaduzzaman and M. M. Rahman, "An adversarial approach for intrusion detection using hybrid deep learning model," in *Proc. Int. Conf. Inf. Technol. Res. Innov. (ICITRI)*, Jakarta, Indonesia, Nov. 2022, pp. 18–23, doi: 10.1109/ICITRI56423.2022.9970221.

[38] C. H. Tseng, W.-J. Tsaur, and Mujiono, "Fuzzy C-means based feature selection mechanism for wireless intrusion detection," in *Proc. Int. Conf. Secur. Inf. Technol. AI, Internet Comput. Big-data Appl.*, Nov. 2022, pp. 143–152, doi: 10.1007/978-3-031-05491-4_15.

[39] Q. Liu, D. Wang, Y. Jia, S. Luo, and C. Wang, "A multi-task based deep learning approach for intrusion detection," *Knowl.-Based Syst.*, vol. 238, Feb. 2022, Art. no. 107852, doi: 10.1016/j.knosys.2021.107852.

[40] W. Wang, S. Jian, Y. Tan, Q. Wu, and C. Huang, "Representation learning-based network intrusion detection system by capturing explicit and implicit feature interactions," *Comput. Secur.*, vol. 112, Jan. 2022, Art. no. 102537, doi: 10.1016/j.cose.2021.102537.

[41] E. Chatzoglou, G. Kambourakis, C. Kolias, and C. Smiliotopoulos, "Pick quality over quantity: Expert feature selection and data preprocessing for 802.11 intrusion detection systems," *IEEE Access*, vol. 10, pp. 64761–64784, 2022, doi: 10.1109/ACCESS.2022.3183597.

[42] S. Ghosh, A. S. M. M. Jameel, and A. E. Gamal, "Improving transferability of network intrusion detection in a federated learning setup," 2024, *arXiv:2401.03560*.

[43] F. Marchiori and M. Conti, "CANEDERLI: On the impact of adversarial training and transferability on CAN intrusion detection systems," in *Proc. ACM Workshop Wireless Secur. Mach. Learn.*, May 2024, pp. 8–13, doi: 10.1145/3649403.3656486.

[44] P. Liao, J. Yan, J. M. Sellier, and Y. Zhang, "Divergence-based transferability analysis for self-adaptive smart grid intrusion detection with transfer learning," *IEEE Access*, vol. 10, pp. 68807–68818, 2022, doi: 10.1109/ACCESS.2022.3186328.

[45] R. Duan, W. Zhao, Z. J. Luo, N. Wang, Y. Liu, and Z. Lu, "Understanding the ineffectiveness of the transfer attack in intrusion detection system," in *Network Security Empowered by Artificial Intelligence* (Advances in information security), 2024, pp. 99–119, doi: 10.1007/978-3-031-53510-9_4.

[46] C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis, "Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 184–208, 1st Quart., 2016, doi: 10.1109/COMST.2015.2402161.

[47] E. Chatzoglou, G. Kambourakis, and C. Kolias, "Empirical evaluation of attacks against IEEE 802.11 enterprise networks: The AWID3 dataset," *IEEE Access*, vol. 9, pp. 34188–34205, 2021, doi: 10.1109/ACCESS.2021.3061609.
[48] S. B. Driss, M. Soua, R. Kachouri, and M. Akil, "A comparison study between MLP and convolutional neural network models for character recognition," *Proc. SPIE*, vol. 10223, May 2017, Art. no. 1022306, doi: 10.1117/12.2262589.

SAUD YONBAWI received the Ph.D. degree in computer science from the University of York, U.K., in 2021. He is currently an Assistant Professor with the College of Computer Science and Engineering, University of Jeddah. His research interests include artificial intelligence and software engineering.

ADIL AFZAL received the M.Phil. degree in computer science from NCBAE University, Lahore, Pakistan, in 2020. His journey in the realm of artificial intelligence (AI) has been shaped by a passion for real-time applications and cutting-edge advancements. As the Founder of XeroAI, he takes pride in leading a team of skilled professionals dedicated to pushing the boundaries of AI innovation. His focus on harnessing the potential of AI technologies aligns seamlessly with his research interests, which include exploring AI's real-time applications and its transformative impact across various sectors. This paper delves into the intricate intersections of AI with practical scenarios, highlighting its role in driving impactful solutions and shaping the future of technology.

MUHAMMAD YASIR received the B.S. degree in information technology from the University of Agriculture Faisalabad, Faisalabad, in 2021. He is currently pursuing the M.Sc.Cs. degree with the University of Engineering and Technology Lahore. From February 2022 to August 2022, he was a Research Officer with the University of Engineering and Technology Lahore. His research interests include intrusion detection systems, cybersecurity, and network security.

MUHAMMAD RIZWAN received the M.Sc. degree from PUCIT, Lahore, Pakistan, in 2006, the M.S. degree from CIIT, Lahore, in 2012, and the Ph.D. degree from HUST, Wuhan, China, in 2017. In 2017, he joined the Department of Computer Science, Kinnaird College for Woman, Lahore, as an Assistant Professor. In 2022, he joined the WMG, University of Warwick, U.K., as an Assistant Professor of cyber security. He is currently an Assistant Professor (U.K. Lecturer) in computer science with the College of Science and Engineering, University of Derby, U.K. He has authored or co-authored several peer-reviewed articles in professional journals and the proceedings of conferences. His current research interests include artificial intelligence, cyber security, e-healthcare, machine learning for cyber security, and deep learning solutions, with a special focus on artificial intelligence-based solutions for telemedicine, smart hospitals, and security management.

NATALIA KRYVINSKA received the Ph.D. degree in electrical and IT engineering from Vienna University of Technology, Austria, and the Habilitation (Docent Title) degree in management information systems from Comenius University Bratislava, Bratislava, Slovakia. She got her Professor title and was appointed for the professorship by the President of the Slovak Republic. She is currently a Full Professor and the Head of the Information Systems Department, Faculty of Management, Comenius University Bratislava. Previously, she was a University Lecturer and a Senior Researcher with the e-Business Department, School of Business Economics and Statistics, University of Vienna. Her current research interests include complex service systems engineering, service analytics, and applied mathematics.