IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 February 2024, accepted 18 March 2024, date of publication 21 March 2024, date of current version 18 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3380014

RESEARCH ARTICLE

Intrusion Detection System for Wireless Sensor Networks: A Machine Learning Based Approach

HALIMA SADIA[1], SAIMA FARHAN[1], YASIN UL HAQ[2], RABIA SANA[2], TARIQ MAHMOOD[3,4], SAEED ALI OMER BAHAJ[5,6], AND AMJAD REHMAN KHAN[3], (Senior Member, IEEE)

[1]Department of Computer Science, Lahore College for Women University, Lahore 54000, Pakistan
[2]Department of Computer Science and Engineering, University of Engineering and Technology Lahore Narowal Campus, Narowal 51600, Pakistan
[3]Artificial Intelligence and Data Analytics (AIDA) Laboratory, College of Computer and Information Sciences (CCIS), Prince Sultan University, Riyadh 11586, Saudi Arabia
[4]Faculty of Information Sciences, University of Education, Vehari Campus, Vehari 61100, Pakistan
[5]MIS Department College of Business Administration, Prince Sattam Bin Abdulaziz University, AlKharj 11942, Saudi Arabia
[6]Department of Computer Engineering, College of Engineering and Petroleum, Hadhramout University, Mukalla, Hadhramout 50511, Yemen

Corresponding author: Saeed Ali Omer Bahaj (saobahaj@gmail.com)

This work was supported by the Higher Education Commission of Pakistan through National Research Program for Universities (NRPU) under Project 17006.

ABSTRACT In this era, plenty of wireless devices are being used with the support of WI-FI (Wireless Fidelity) and need to be maintained and authorized. Wireless Sensor Networks (WSN), a cornerstone of modern wireless technology, offer cost-efficient solutions for diverse monitoring tasks but are exposed to many security threats, including unauthorized access, attacks, and suspicious activities. These vulnerabilities can significantly degrade the performance and reliability of WSNs, making the early detection and mitigation of such threats imperative. Intrusion Detection Systems (IDS) are crucial tools in safeguarding WSNs against these challenges. Numerous studies focus on enhanced Intrusion Detection model accuracy and decrease in loss with higher Detection Rate and lower False Alarm Rate, because of this, eliminating the repetitive feature of the dataset is exhibited. This study introduces a sophisticated Network Intrusion Detection System (NIDS) to safeguard Wi-Fi-based WSNs from prevalent cyber threats, such as impersonation, flooding, and injection attacks. At the heart of our approach is a meticulous feature selection process that enhances the dataset's utility by eliminating null values, substituting unknown entries with a placeholder ('NONE'), and refining the feature set to include only the most relevant indicators of potential security breaches. Initially, from a pool of 154 features, a subset of 76 is selected, further narrowed down to 13 pivotal features, ensuring a focused and efficient analysis. Employing standard scaler function for feature scaling and preprocessing, this research train proposed a Convolutional Neural Network (CNN) based approach aiming for optimal intrusion detection and prevention across multiclass classifications within WSN environments. The study aims to enhance detection accuracy, reduce loss values, and decrease false alarm rates, comparing it to CNN, Deep Neural Network (DNN) (5), DNN (3), and (Long Short-Term Memory) LSTM networks. The model's performance is evaluated using various metrics, including precision, recall, support, F1 score, and macro-average. The culmination of our research efforts is evidenced by the exceptional performance of the CNN model, achieving an impressive accuracy rate of 97% and a loss metric of 0.14, all while maintaining a minimal False Alarm Rate. This study significantly advances IDS accuracy while simultaneously reducing false alarms, thus fortifying the security posture of WSNs in the face of evolving cyber threats.

INDEX TERMS WSN, Wi-Fi, NIDS, WIDS attacks, security issues, network threats, feature engineering, multiclass classification, inclusive innovations.

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

I. INTRODUCTION

Wireless Sensor Network (WSN) is one of the ultimate sources for communication of wireless base network

domain [1]. WSNs integrate with the help of sensor nodes that establish with different topologies such as star, tree, or mesh [2], it senses the data flow and transmission within the wireless network, and the dominant service-abilities of these nodes are sensing, processing, computation, and communication [3]. WSNs are used for monitoring, protecting, and tracking purposes that provide the most cost-effective platform with fewer energy resources for wireless network traffic communication [4]. WSNs can be prone to unauthorized access from within or out of the network by any intruders therefore, effective measures for the protection of sensor nodes are done to maintain the network security as well [5].

WIFI based sensors are broadly accessible on the wireless network and on the wireless infrastructure, and their data is reachable anywhere in the world by TCP/IP via computers or smartphones. Repeaters can be added as access points if the sensor is not in the range of the WI-FI access point. Sensors can join a network by knowing SSID and passphrase and, with the help of a URL address or IP address, easily send data to the server as well [6].

An Intrusion Detection System (IDS) is a Software or Hardware system implemented to detect and prevent unauthorized access to any computer system or network [7]. IDS is in two types: host-based, integrated on a host device and checks process and user activity on the local machine to detect intrusion or network-based. The most commonly used IDS is established over a network and works within a network system in a distributed manner to check traffic flow for intrusions [8], [9]. IDS classifies Aegean Wi-Fi Intrusion Dataset (AWID) into the most commonly four types of classes: Normal, Flooding, Injection, and Impersonation, according to their behavior on the wireless network. IDS attack detection can be distributed into four types: Anomaly-based IDS, suitable for unknown attacks detection that triggers abnormal behavior; misuse-based or signature IDS, ideal for detection of known attacks verified based on the predefined signature [10], [11], specification-based IDS detects abnormality for network components like routing tables, nodes and protocols by a set of rules and thresholds. Hybrid-based IDS is the combination of anomaly, signature, and specification-based IDS [12]. Wireless networks have gained a high degree of strength over wired networks [13] because of their flexibility, high accessibility, mobility, and no need for any extra infrastructure. Wireless Intrusion Detection Systems (WIDS) are sensors designed to monitor network intrusion, protecting WLANs 24/7 by listening to all inside operating frequencies of WLANs.

Many machine learning and deep learning techniques, like Support Vector Machine (SVM), Perceptron, K Nearest Neighbor (KNN), Feed Forward Neural Networks (FFNN), Deep Neural network (DNN) and Convolutional Neural Networks (CNN) are getting hype while implementing a Network IDS for WI-FI-based WSNs. This research proposes deep learning and machine learning-based Intrusion Detection for WI-FI networks as having a pre-processed feature set. As a result, an accurate, intrusion, and minimum loss value network will be achieved.

This proposed work includes a combination of Deep Learning DNN, Recurrent Neural Networks (RNN-LSTM), CNN models, and Machine Learning LR using WI-FI-based WSN, effective for the 802.11 network threat identification and classification, and AWID for Network Intrusion Detection System. This Network-IDS is used for multiclass and binary classifications with the feature set of 76 and 13.

A. PROBLEM STATEMENT

The Internet of Things (IoT) security presents significant threats to wireless networks, particularly in identifying and neutralizing malicious activities. Rapid and accurate detection of unauthorized or irregular network traffic is crucial for detecting potential intrusion efforts or attacks. Advanced Intrusion Detection Systems (IDS) are needed to differentiate between benign and nefarious network behaviors. Efficient IDS efficiency is essential for immediate detection and intervention, especially in fast-paced, resource-constrained wireless networks. Unique threats, such as Flooding, Injection, and Impersonation attacks, require bespoke detection and countermeasure strategies. This investigation aims to contribute to the evolution of IDS technologies, fortifying wireless network defenses against sophisticated threats. The goal is to navigate the complexities of wireless network security and pave the way for future advancements in IDS capabilities, enhancing our collective ability to protect against the ever-changing cybersecurity landscape.

B. CONTRIBUTION

DNNs are a promising solution for intrusion detection in IoT security due to their ability to identify and classify anomalous data. IoT devices generate large volumes of data, but any disruption disrupts the typical pattern, making it an outlier. This research proposes a communication protocol-independent DNN-based real-time IoT intrusion detection system for safer and more secure IoT environments. The proposed system does not require system attributes, communication protocol adjustment, or network structure virtualization. It detects intrusions based on the deviation from the regular signal pattern, making it a plug-and-play solution for IoT security. This research has the following major contributions:

- Identifying Key Features of Datasets for Intrusion Detection in IoT Networks
- The study uses advanced feature selection and scaling techniques to optimize classification performance by focusing on the most discriminative features, resulting in feature sets of 76 and 13 dimensions.
- To improve model efficiency by converting multi-class AWID data into binary labels, streamlining the classification task, and enhancing model training and evaluation.
- The study examines the effectiveness of different neural network architectures, including DNN (5), DNN (3),

CNN, and RNN-LSTM, in binary and multiclass classification tasks, highlighting their strengths and weaknesses in WSN-based IDS.
- An averaging test is designed by combining all fitted models DNN (5), DNN (3), CNN, and RNN-LSTM with a list to test the predictions using any single array from the test dataset, enhancing prediction accuracy and reliability by leveraging the collective insights of multiple models.
- The study assesses the performance of all designed models using various metrics such as Detection Rate, False Alarm Rate, F1-measure, Support, Micro Average, Macro Average, Accuracy, Recall, and Precision, providing a detailed understanding of each model's strengths and weaknesses.

II. LITERATURE REVIEW

In this section, various approaches to Machine Learning and Deep Learning, feature sets, and results using AWID are to be discussed in Table 1 and 2.

A. MACHINE LEARNING BASED APPROACHES

Since wireless networks are getting a lot of hype, Kolias et al. [14] created a publicly available AWID for IDS by analyzing 'intruder' and 'normal' traffic over 802.11 networks. They also detected different 'intruder' types and their specific patterns using machine learning approaches like Naöve Bayes, AdaBoost, Hyperpipes, J48, OneR, Random Forest, Random Tree, and ZeroR. J48 was the best performed on both feature sets of 156 and 20.

Machine Learning models are time-consuming in the training phase so the reduced set of attributes is really helpful to speed up the training [8], [15]. Keep this thing in count: Bhandari et al. [17] presented the Shapley Additive Explanations (SHAP) technique for feature reduction based on Tree-based classifiers that were CatBoost, Random Forest, LightGBM, and XGBoost. SHAP selects 15 features of AWID that were most helpful for classification. There was no big difference in accuracy values but SHAP improved training time well. IDS systems are also part of smart cities, so to predict the attack for such a system, Gaber et al. [18] selected recursive-based feature elimination and constant removal techniques to preprocess the AWID. Furthermore, this processed data is used to classify the ''injection'' class by Support Vector Machine (SVM), Random Forest (RF), and Decision Tree (DT) with the Feature Set (FS) of 76, 13, and 8. After implementing all these experiments, DT-based classification with FS of 8 achieved the highest accuracy of 99% [10], [20]

To make the Intrusion detection system more effective, feature engineering is considered a promising approach. Thanthrige et al. [21] proposed how feature reduction techniques, information gain, and Chi-squared statistics helped in detection accuracy and classification speed. Experiments for Intrusion detection were performed on three feature sets of 111, 41, and 10 using machine learning models like Random Tree, OneR, Random Forest, AdaBoost, and J48, and all of these algorithms gained an accuracy above 90%.

Using a combinational technique for feature extraction and selection is more effective than any single technique. Rahman et al. [22] proved this scenario productively. This research work used the combination of C4.8, SVM, and NB for wrapper-based feature selection. Next, ANN was applied for classifying the ''normal'' and ''impersonation'' classes using 20 features, which resulted in 99.95%. Intrusion detection is a common issue when a network crashes to provide adequate security to an application [23]. To prevent this, the network should have the necessary features. Gavel et al. [24] proposed an efficient method to design a defense mechanism that considers the various factors that affect the correlation between features and their allocated rank. The useful features are then extricated using the OMCFR (Optimized Maximum Correlation Feature Reduction). The FS of 140 got after this and applied to Random Forest for AWID-based IDS, resulting in an accuracy of 99.2

Park et al. [25] introduce G-IDCS, a graph-based intrusion detection and classification system, to improve in-vehicle network security. It overcomes the limitations of existing intrusion detection systems, such as requiring large CAN messages and not classifying attack types. The threshold-based method reduces detection time by over 1/30 and improves combined attack detection accuracy by over 9%. The machine learning-based attack-type classifier outperforms existing techniques. Kandhro et al. [26] developed a deep learning-based method for intrusion detection in IoT-driven IIC networks, overcoming issues like poor accuracy, ineffectiveness, high false positives, and inability to handle new intrusion types. Using a generative adversarial network, the framework improved accuracy, reliability, and efficiency by 95% to 97%, outperforming state-of-the-art DL classifiers. Boahen et al. [27] developed OPT_NSDAE, a deep learning method for unsupervised feature learning, which effectively detects compromised accounts on Online Social Networks (OSN) by reducing human interaction in feature selection and extraction, outperforming traditional schemes.

B. DEEP LEARNING BASED APPROACHES

Wireless technologies are way more rapid for a large amount of data exchange, and IDS helps to secure these technology networks. DNN-based wireless IDS was proposed by Kasongo et al. [35] in which a feature set of 26 was used and extracted by a wrapper-based feature extraction unit (WFEU). In this research work, feed-forward DNN was used for binary and multiclass classification, and the obtained results were 99.66% and 99.77%, respectively. This work's accuracy is higher than all of the compared ML models: DT, RF, NB, kNN, and SVM [36].

AWID dataset mainly contains 16 classes of intrusion and IDS is used to detect malicious traffic over the network. Aminanto et al. [37] proposed a fully unsupervised k-mean

TABLE 1. Comparison of existing machine learning based models using AWID.

Ref.	Feature Set	Approach	Classification
[28]	154	AdaBoost	Multiclass
	20	J48	
		OneR	
		Random Forest	
		Random Tree	
		HYperpipes	
		ZeroR	
		Naive Bayes	
[21]	111	OneR	Multiclass
	41	Random Forest	
	10	Random Tree	
		AdaBoost	
		J48	
[29]	5	Linear Support Vector Machine	Impersonation class
[17]	15	RF	Multiclass
		XGBoost	
		LGBM	
		CatBoost	
[30]	34	Bagging	Multiclass
		RF	
		ET	
		XGBoost	
		NB	
[22]	20	ANN	Binary
[31]	8	Self-adaptive grasshopper algorithm	Multiclass
[18]	76	Decision Tree	Injection class
	13	SVM	
	8	Random Forest	
[24]	140	Radom Forest	Binary
[32]	154	Cascaded SVM	Multiclass
[33]	25	Supervised Clustering Based Classifier	Multiclass
		KNN	
		SVM	
[34]	37	Random Forest	Binary
		Bagging	
		Extra Tree	
		XGBoost	

clustering machine learning approach using 50 features extracted by a stacked auto-encoder for the ''impersonation'' attack class. This approach does not need any prior labeling of the dataset during the training phase and resulted in a 92% detection rate [38].

For any IDS feature engineering, feature extraction and feature selection are important before applying any classification algorithm. Kim et al. [39] performed feature engineering with SAE using two hidden layers having 100, and 50 neurons. FS from SAE fed to the deep k-mean clustering algorithm applied for classifying ''normal'' and ''impersonation'' class with k=2, resulted in an accuracy of 94.81%.

To improve the IDS, Wang et al. [40] presented deep learning-based approaches such as SAE, DNN having 3 layers, and DNN having 7 layers for classification using a feature set of 71 produced after preprocessing. As a result, the highest numbers of accuracy achieved were from 7 layered DNN for ''normal'', ''impersonation'', ''injection'', and ''flooding'' were 98.46%, 99.99%, and 98.39% and 73.12%, respectively.

TABLE 2. Comparison of existing deep learning based models using AWID.

Ref.	Feature Set	Approach	Classification
[42]	154	Stacked auto-encoder	Impersonation class
	35		
[37]	50	Unsupervised k-mean clustering	Impersonation class
[39]	154	Deep k-mean clustering	Binary
[40]	71	Stacked auto-encoder	Multiclass
		DNN 3 layered	
		DNN 7 layered	
[41]	154	Deep Reinforcement Learning	Multiclass
		DQN	
		DDQN	
		PG	
		AC	
[35]	26	Feed Forward Deep Neural Network	Multiclass Binary
[43]	154	Sparse Autoencoder with swish-PReLU	Binary
[44]	35	Auto-encoder DNN	Multiclass
[45]	154	Auto encoder	Multiclass
		Supervised Clustering model	
		MLP	
[46]	74	CNN	Multiclass Binary
[5]	23	DNN	Multiclass

Intrusion Detection Systems play crucial roles in the increase in network technology and its associated threats. Time-to-time improvement of IDS is also top of the discussion. Martin et al. [41] presented DRL DQN, DDQN, PG, and AC algorithms. After implementing all DRL algorithms, the result from Double Deep Q-Network was the best. The one-Vs-Rest technique was also applied to DDQN, which showed the ''impersonation'' attack class had the poorest performance out of all classes.

III. MATERIALS AND METHODS

A. DATASET

Several datasets, KDD Cup 99, NSL-KDD, and UNSW-NB15 have been publicly available and used to detect intrusion of network-based systems [20]. The Aegean Wi-Fi Intrusion Detection (AWID) is the only publicly accessible dataset at (https://icsdweb.aegean.gr/awid/download-dataset) produced in 2015 with real and immense WI-FI network traces for wireless-based sensor network systems that going to be used in this research work. AWID is naturally constructed from a Wireless Local Area Network (W-LAN) instead of artificial generation and linked by Wired Equivalent Protocol (WEP) [8].

AWID contains two dataset groups, with a further two subgroups of the Training (Trn) and Testing (Tst) datasets.

- Reduced group (AWID-CLS-R-Trn), (AWID-CLS-R-Tst), (AWID-ATK-R-Trn), (AWID-ATK-R-Tst).
- Full group (AWID-CLS-F-Trn), (AWID-CLS-F-Tst), (AWID-ATK-F-Trn), (AWID-ATK-F-Tst).

AWID reduced group (AWID-CLS-R-Trn), (AWID-CLS-R-Tst), (AWID-ATK-R-Trn), (AWID-ATK-R-Tst) is going to be used in this research, having 1,795,575 training and

575,643 testing records. The dataset's Comma Separated Values (CSV) file contains 935 MB of hard disk space.

AWID is made for Wireless Personal Area Networks with the help of multiple wireless devices [47]. One and all packets of the dataset constitute the vector of 155 attributes with 154 input features shown in Table 3, which stores MAC Layer information and the class.

AWID contains four types of classes, of which three are ''intruder'' classes.

- Normal
- Flooding
- Injection
- Impersonation

Data distribution of AWID in terms of attack can be seen in Table 4.

B. TOOLS

1) GOOGLE COLAB

Google Colab is a cloud-based tool that is crucial in developing machine learning-based IDS for WSNs. It offers a cost-free environment, allowing for efficient processing of large datasets and complex algorithms. The platform's user-friendly interface and seamless Google Drive integration foster collaboration and real-time sharing. Its compatibility with leading data science libraries like TensorFlow, PyTorch, and Keras enhances IDS accuracy and efficiency. Google Colab's scalability ensures computational resources can meet project demands, supporting innovation in IDS methodologies without hardware constraints. Overall, Google Colab significantly aids in cybersecurity development and collaboration. It has plenty of advantages and features. Some of them are as follows:

- No need to install any setup in your local space.
- Have preinstalled libraries.
- Can create, upload, and share notebooks.
- Can import and save the notebook from Google Drive.
- Can import and publish datasets and codes directly from any external sources such as GitHub, Kaggle, etc.
- Can integrate many open-source frameworks like TensorFlow, PyTorch, and OpenCV.
- Can facilitate with the GPU, TPU, and free cloud services.
- Supports the code having mathematical equations.

2) PANDAS

Pandas is a Python library crucial in creating IDS for WSN using ML. It efficiently manages large data sets, streamlines data preprocessing, and supports intricate feature engineering for anomaly identification. Pandas' integration with visualization tools enhances data trends and irregularities, essential for monitoring WSNs. Its strength in handling time-series data is particularly beneficial in WSNs, where data is often time-dependent. Pandas' seamless compatibility with other Python-based ML libraries creates a cohesive development environment, making it essential to designing effective and sophisticated ML-driven IDS for WSNs. [48].

3) NUMPY

NumPy is a crucial library for IDS for WSN using a Machine Learning-based approach. Its ability to handle large, multi-dimensional arrays at high speeds is essential for processing and analyzing complex sensor data from WSNs. Thanks to its C/C++ implementation, NumPy's computational efficiency enables faster data manipulation and transformation, which is essential for training and deploying machine learning models for IDS. Its array operations are ideal for intensive numerical computations, making it a valuable tool in data science, signal processing, and image processing.

4) MATOPLOTLIB

Matplotlib is a valuable tool for data visualization in an IDS for WSN using Machine Learning. It is a cross-platform plotting library for Python that effectively represents complex datasets, aiding in understanding patterns and anomalies in WSN data. Its integration with Python makes it a practical alternative to MATLAB, especially when used alongside libraries like NumPy. Written in multiple languages, Matplotlib helps develop and evaluate effective IDS strategies.

5) TENSORFLOW

TensorFlow, an open-source library for machine and deep learning, is crucial in developing IDS for WSN. It simplifies complex algorithm implementations and is compatible with Python. TensorFlow version 1.15.2 is the backend in this research, providing powerful tools for processing vast data from WSNs and addressing network security challenges.

6) KERAS

Keras is a Python-based neural network library suitable for IDS for WSN. It supports multiple backends like TensorFlow, Theano, and MXNet for neural network computations. Its simplicity makes it an excellent choice for beginners. Keras's Python-based nature enhances its utility in IDS applications for WSNs, making it an accessible and powerful tool for complex neural network architectures.

7) SKLEARN

Scikit-learn is a Python-based machine-learning library used for developing IDS for WSN. It offers a range of statistical, mathematical, and general-purpose algorithms, making it a robust tool for modeling IDS. It integrates seamlessly with foundational libraries like NumPy, Pandas, and Matplotlib, facilitating feature selection and model building for the accurate detection of anomalies and security threats. It supports various classification, regression, and clustering techniques, and its tools for data preprocessing and model evaluation are invaluable for real-world applications.

TABLE 3. Dataset description.

Feature no.	Feature Name	Feature no.	Feature Name
f1	frame.interface_id	f79	wlan.sa
f2	frame.dlt	f80	wlan.bssid
f3	frame.offset_shift	f81	wlan.frag
f4	frame.time_epoch	f82	wlan.seq
f5	frame.time_delta	f83	wlan.bar.type
f6	frame.time_delta_displayed	f84	wlan.ba.control.ackpolicy
f7	frame.time_relative	f85	wlan.ba.control.multitid
f8	frame.len	f86	wlan.ba.control.cbitmap
f9	frame.cap_len	f87	wlan.bar.compressed.tidinfo
f10	frame.marked	f88	wlan.ba.bm
f11	frame.ignored	f89	wlan.fcs_good
f12	radiotap.version	f90	wlan_mgt.fixed.capabilities.ess
f13	radiotap.pad	f91	wlan_mgt.fixed.capabilities.ibss
f14	radiotap.length	f92	wlan_mgt.fixed.capabilities. cfpoll.ap
f15	radiotap.present.tsft	f93	wlan_mgt.fixed.capabilities.privacy
f16	radiotap.present.flags	f94	wlan_mgt.fixed.capabilities.preamble
f17	radiotap.present.rate	f95	wlan_mgt.fixed.capabilities.pbcc
f18	radiotap.present.channel	f96	wlan_mgt.fixed.capabilities.agility
f19	radiotap.present.fhss	f97	wlan_mgt.fixed.capabilities.spec_man
f20	radiotap.present.dbm_antsignal	f98	wlan_mgt.fixed.capabilities.short_slot_time
f21	radiotap.present.dbm_antnoise	f99	wlan_mgt.fixed.capabilities.apsd
f22	radiotap.present.lock_quality	f100	wlan_mgt.fixed.capabilities.radio_measurement
f23	radiotap.present.tx_attenuation	f101	wlan_mgt.fixed.capabilities.dsss_ofdm
f24	radiotap.present.db_tx_attenuation	f102	wlan_mgt.fixed.capabilities.del_blk_ack
f25	radiotap.present.dbm_tx_power	f103	wlan_mgt.fixed.capabilities.imm_blk_ack
f26	radiotap.present.antenna	f104	wlan_mgt.fixed.listen_ival
f27	radiotap.present.db_antsignal	f105	wlan_mgt.fixed.current_ap
f28	radiotap.present.db_antnoise	f106	wlan_mgt.fixed.status_code
f29	radiotap.present.rxflags	f107	wlan_mgt.fixed.timestamp
f30	radiotap.present.xchannel	f108	wlan_mgt.fixed.beacon
f31	radiotap.present.mcs	f109	wlan_mgt.fixed.aid
f32	radiotap.present.ampdu	f110	wlan_mgt.fixed.reason_code
f33	radiotap.present.vht	f111	wlan_mgt.fixed.auth.alg
f34	radiotap.present.reserved	f112	wlan_mgt.fixed.auth_seq
f35	radiotap.present.rtap_ns	f113	wlan_mgt.fixed.category_code
f36	radiotap.present.vendor_ns	f114	wlan_mgt.fixed.htact
f37	radiotap.present.ext	f115	wlan_mgt.fixed.chanwidth
f38	radiotap.mactime	f116	wlan_mgt.fixed.fragment
f39	radiotap.flags.cfp	f117	wlan_mgt.fixed.sequence
f40	radiotap.flags.preamble	f118	wlan_mgt.tagged.all
f41	radiotap.flags.wep	f119	wlan_mgt.ssid
f42	radiotap.flags.frag	f120	wlan_mgt.ds.current_channel
f43	radiotap.flags.fcs	f121	wlan_mgt.tim.dtim_count
f44	radiotap.flags.datapad	f122	wlan_mgt.tim.dtim_period
f45	radiotap.flags.badfcs	f123	wlan_mgt.tim.bmapctl.multicast
f46	radiotap.flags.shortgi	f124	wlan_mgt.tim.bmapctl.offset
f47	radiotap.datarate	f125	wlan_mgt.country_info.environment
f48	radiotap.channel.freq	f126	wlan_mgt.rsn.version
f49	radiotap.channel.type.turbo	f127	wlan_mgt.rsn.gcs.type
f50	fradiotap.channel.type.cck	f128	wlan_mgt.rsn.pcs.count
f51	radiotap.channel.type.ofdm	f129	wlan_mgt.rsn.akms.count
f52	radiotap.channel.type.2ghz	f130	wlan_mgt.rsn.akms.type
f53	radiotap.channel.type.5ghz	f131	wlan_mgt.rsn.capabilities.preauth
f54	radiotap.channel.type.passive	f132	wlan_mgt.rsn.capabilities.no_pairwise
f55	radiotap.channel.type.dynamic	f133	wlan_mgt.rsn.capabilities.ptksa_replay_counter
f56	radiotap.channel.type.gfsk	f134	wlan_mgt.rsn.capabilities.gtksa_replay_counter
f57	radiotap.channel.type.gsm	f135	wlan_mgt.rsn.capabilities.mfpr
f58	radiotap.channel.type.sturbo	f136	wlan_mgt.rsn.capabilities.mfpc
f59	radiotap.channel.type.half	f137	wlan_mgt.rsn.capabilities.peerkey
f60	radiotap.channel.type.quarter	f138	wlan_mgt.tcprep.trsmt_pow
f61	radiotap.dbm_antsignal	f139	wlan_mgt.tcprep.link_mrg
f62	radiotap.antenna	f140	wlan.wep.iv
f63	radiotap.rxflags.badplcp	f141	wlan.wep.key
f64	wlan.fc.type_subtype	f142	wlan.wep.icv
f65	wlan.fc.version	f143	wlan.tkip.extiv
f66	wlan.fc.type	f144	wlan.ccmp.extiv
f67	wlan.fc.subtype	f145	wlan.qos.tid
f68	wlan.fc.ds	f146	wlan.qos.priority
f69	wlan.fc.frag	f147	wlan.qos.eosp
f70	wlan.fc.retry	f148	wlan.qos.ack
f71	wlan.fc.pwrmgt	f149	wlan.qos.amsdupresent
f72	wlan.fc.moredata	f150	wlan.qos.buf_state_indicated
f73	wlan.fc.protected	f151	wlan.qos.bit4
f74	wlan.fc.order	f152	wlan.qos.txop_dur_req
f75	wlan.duration	f153	wlan.qos.buf_state_indicated
f76	wlan.ra	f154	data.len
f77	wlan.da	f155	class
f78	wlan.ta		

TABLE 4. AWID classes distribution.

	Normal	Flooding	Injection	Impersonation	Total
AWID-CLS-R-Trn	1,633,190	48,484	65,379	48,522	1,795,575
AWID-CLS-R-Tst	530,785	8,097	16,682	20,079	575,643
Total	2,163,975	56,581	82,061	68,601	2,371,218

C. PROPOSED METHOD

Our research is devoted to developing an IDS for WSNs, with a specific focus on utilizing Wi-Fi frames from network sensors to detect various types of intrusions. Our methodology's cornerstone is applying deep learning techniques, specifically the utilization of CNNs. We have chosen CNNs due to their exceptional feature extraction and pattern recognition capabilities, which are vital for accurately identifying intrusions in network traffic. The high-dimensional data characteristic of WSNs is well-managed by CNNs. Furthermore, we have selected a comprehensive feature set comprising 76 attributes to cater to binary and multiclass classification, ensuring a wide-ranging approach to various intrusion scenarios. To provide a robust scientific basis for our methodology, we have also evaluated other models, such as SVMs and traditional machine learning techniques. However, these were found to be less effective than CNNs, particularly in deep feature extraction and handling the dynamic nature of WSN traffic. Our decision-making process is grounded in scientific principles, focusing on the necessity for models that efficiently process complex, high-dimensional data. This approach transforms our methodology from a procedural description into a scientifically exploratory narrative. Our proposed method involves three key phases: preprocessing, feature engineering, and classification. We employ an embedded feature selection method integrated into the model training process, identifying the prediction variable's most contributive features. This indicates that feature selection is intrinsically linked to model training, enhancing the model's performance and accuracy. The primary steps of our model include preprocessing the AWID dataset, performing feature selection and reduction, employing feature engineering, splitting the dataset for classification, selecting appropriate deep learning and machine learning models for intrusion detection, and evaluating the results for continuous improvement. This embedded feature selection strategy is a critical component in enhancing the performance and accuracy of our model.

The main steps of the proposed model are shown in Figure 1, and as follows:

- Preprocessing steps are applied to the AWID dataset. In the next step, feature selection and reduction are performed.
- Feature engineering helps in feature selection and reduction.
- Splitting the dataset for classification.
- Deep Learning and Machine Learning Models are selected for Intrusion Detection.
- Results.
- Evaluation for improving our results.

1) DATA PREPROCESSING AND FEATURE ENGINEERING

The study focuses on developing an IDS for WSN through a data preprocessing process. This process involves transforming raw data from the AWID dataset into a refined format suitable for machine learning models. Key steps in this process include data cleaning, where errors and inconsistencies in the AWID dataset, particularly in the training (AWID-R-Trn) and testing (AWID-R-Tst) subsets, are corrected, as shown in Figure 2. Data integration is also crucial, ensuring that diverse data forms from various sources in WSNs are amalgamated into a consistent dataset. This is followed by data reduction, where the vast amount of data in the AWID dataset is distilled to its most essential elements, thus enhancing data processing efficiency. Finally, data transformation involves converting the data into a format optimal for machine learning, including scaling, normalization, and potential feature engineering. These preprocessing steps ensure the training and testing of machine learning models on high-quality, relevant data, which is essential for effective and accurate intrusion detection in WSNs.

2) DROP NULL VALUES

AWID contains many null values in columns and rows. These values are assigned as ''0'', then drop columns with 50% null values and all null rows.

3) REPLACE "?" WITH "NAN"

AWID many values listed as ''?'' symbol are replaced with ''Nan'', Nan=not a number, to represent missing values.

4) REPLACE "NAN" WITH "0"

Replaced ''Nan'' replaced with '' 0'', then eventually dropped as unnecessary values.

5) CONVERSION OF DATATYPES

There are some of the features having float and hexadecimal datatype values that are converted into integer datatype.

6) STANDARD SCALER

Machine Learning function is used in preprocessing and features scaling of training and testing datasets. Feature scaling is the important step for modeling, done by normalization before training, because it helps to select features with more weight at the time of result and convert the data points into the 0-1 range [3].

7) LABEL ENCODER

Label Encoder is a Sklearn tool used for the normalization of labels. It transforms nonnumerical labels to numerical labels between 0 and n_classes-1 where's n=number of

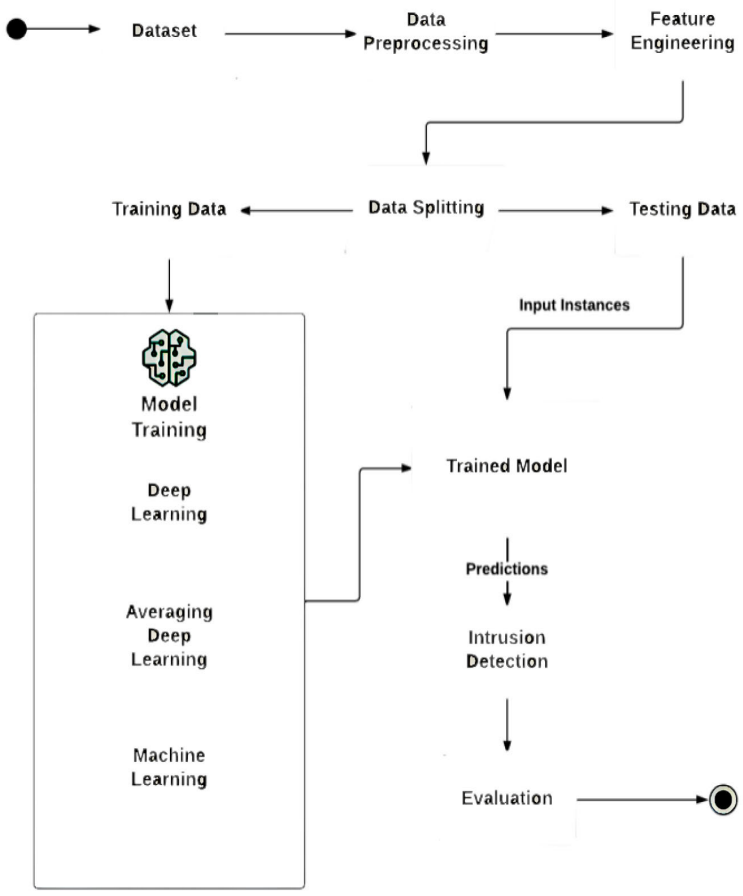


FIGURE 1. Proposed method.

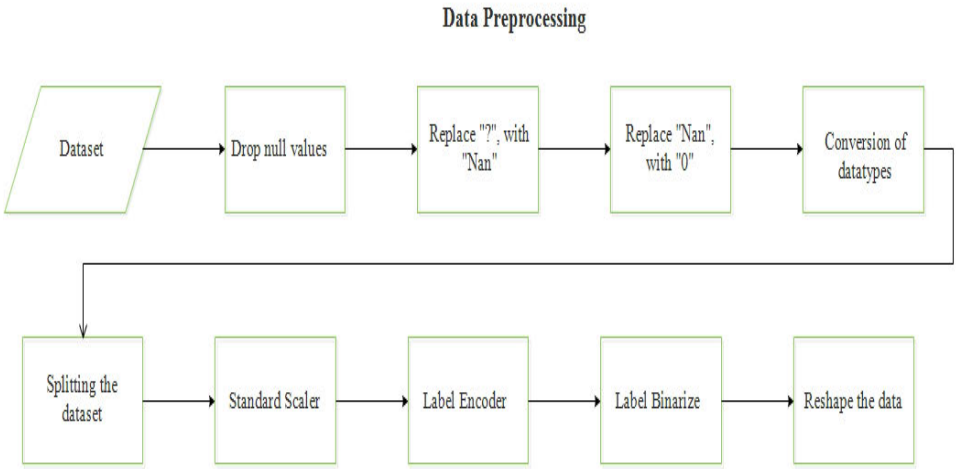


FIGURE 2. Data preprocessing.

distinct labels. This AWID class encodes labels as [0, 1, 2], 0=impersonation, 1=injection, and 2=normal. The output of the label encoder changes into categorical labels of 0,1, 0=False, and 1=True.

8) LABEL BINARIZE

Sklearn has many regression and binary classification algorithms and, by using the one-vs-all technique, converts them into multi-class classification learning algorithms.

9) RESHAPE THE DATA

To work with LSTM and CNN input array reshaped into the 3D array using NumPy.

10) SPLITTING THE DATASET

AWID dataset has two files for training and testing sets and applied validation split to the training set according to the performed algorithm. Feature Engineering is performed for feature selection and extraction. AWID has a lot of redundant values that are not good enough for classification using any machine learning or deep learning algorithm. Preprocessing helped in data cleaning by removing null values of columns and null rows and converting column values into integer data. After these steps, a feature set of 76 is obtained; by applying feature selection on 76 features, we have a feature set of 13. All experiments are performed using both 76 and 13 feature sets.

D. ARCHITECTURE OF PROPOSED METHOD

The architecture of the proposed method of WSN-based IDS is divided into two phases; Deep Learning-based model classification and Machine learning-based model classification. In deep learning-based classification, four models are being designed and implemented for binary classification as well as multiclass classification using DNN with 5 layers, DNN with 3 layers, CNN, and RNN with LSTM having feature sets of 76 and 13, and the result is being concluded by averaging all these fitted models to classify the network traffic effectively. To check probabilities within AWID classes, machine learning-based classification models are being designed and implemented using Logistics Regression (LR) having feature sets of 76 and 13. For binary classification with both feature sets, Binominal and Binominal with c=10.0 are being executed. As LR cannot perform multiclass classification to make it work, One-Vs-Rest and One-Vs-One techniques are applied in which multiclass classification diverges into multiple binary classifications referred to as Multinomial. A detailed view of the proposed model architecture can be seen in Figure 3.

E. DEEP LEARNING MODEL CLASSIFICATION

Deep learning-based Neural Networks made machine learning more intelligent in pattern learning and suggestions. For the implementation of WSN-based IDS using Keras, here are some network layers that are being used in this research work shown in Table 5.

1) INPUT LAYER

This layer contains artificial-based neurons that help to bring initial data into the system and then pass it to the rest of the network for further processing. Every neural network has only 1 input layer and is being added to implement DNN, CNN, and RNN-LSTM.

TABLE 5. Deep learning based models implementation review.

	DNN(5)	DNN(3)	CNN	RNN-LSTM
Input Layer	✓	✓	✓	✓
Dense Layer	✓	✓	✓	X
Dropout Layers	✓	✓	✓	✓
Conv1D	X	X	✓	X
MaxPooling1D	X	X	✓	X
LSTM	X	X	X	✓
Flatten Layer	X	X	✓	✓
Output Layer	✓	✓	✓	✓
ReLU Activation Function	✓	✓	✓	✓
Softmax Activation Function	✓	✓	✓	✓
Adam Optimizer	✓	✓	✓	✓
Loss=categorical_crossentropy	✓	✓	✓	✓
Early Stopping	✓	✓	✓	✓

2) DENSE LAYER

The dense layer is the entry-level layer equipped by Keras. It is the neural network''s most commonly used non-linear regular deep neural layer, in which all neurons of layers receive the required parameters. If it is set as the first layer, then Input Shape is provided; otherwise, it receives input from the previous layer's output. This layer is used for matrix-vector multiplication. The values under the matrix are trained parameters of the previous layer. The more layers are to be added, it will become more complex. Using each neuron of the preceding layer, the dimensions of the vector can be changed. So, the output of this layer is an N-dimensional vector. This layer is being added to the proposed DNN, CNN, and RNN-LSTM models.

3) DROPOUT LAYER

As neural networks are flexible, there are higher chances of overfitting as well as co-adaption, where multiple neurons try to extract similar or the same hidden features from input data to deal with this issue and to save the machine''s resources from wasting the Dropout layer is to be applied by intentionally dropping out neurons by zeroing out the values of the neurons temporarily during model training. This layer is being added to the proposed DNN, CNN, and RNN-LSTM models.

4) CONV1D LAYER

While implementing CNN, the ConviD layer is to be applied to extract features from input data and set padding to ''same'' that fills zeros to both the left and right sides of the input because CNN works only on 3-dimensional array type data.

5) MAXPOOLING1D LAYER

MaxPooling1D is used to reduce the dimensions of feature maps, the number of learning parameters as well as performance computation of the network. It helps to select high-level input features specified by 'pool_size' and calculate the maximum value for each patch of the feature map. This layer is applied while implementing CNN.

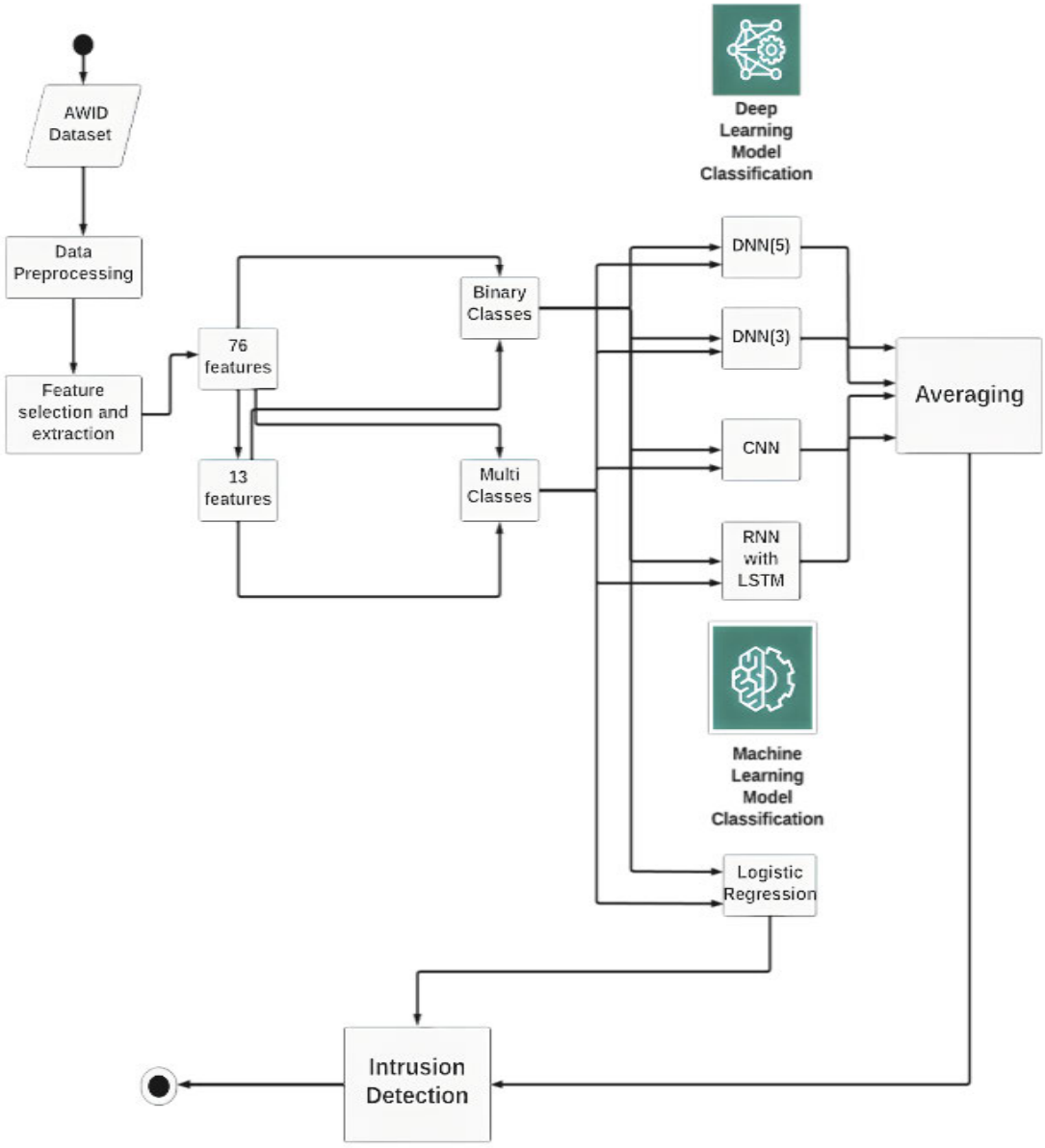


FIGURE 3. Architecture of proposed model.

6) LSTM LAYER

LSTM is used to maximize the performance of RNN by predicting the long sequence of data with 'return_sequences= True'.

7) FLATTEN LAYER

The flattened layer is used to reshape the input size, this layer is being implemented for CNN, RNN-LSTM to convert the 3D array into a 2D array for generating output.

8) OUTPUT LAYER

The output Layer is the final layer used to obtain desired output predictions and the number of outputs to be specified in this layer.

9) RELU ACTIVATION FUNCTION

Rectified Linear Unit is the activation function used in multi-layered neural networks. It boosts the training speed and the computational performance of DL-based neural networks. This function can be defined as given:

$$f(x) = max(0, x)x = inputvalues \tag{1}$$

$$f(x) = \begin{cases} 0, & \text{if } x < 0 \\ x, & \text{if } x \geq 0 \end{cases} \tag{2}$$

10) SOFTMAX ACTIVATION FUNCTION

This function is used in the output layer to make predictions easier to interpret neural networks by transforming the raw output of NN into the vector of probability.

11) ADAM OPTIMIZER

Adaptive Movement Estimation optimizer is to be used as the default optimizer for faster computation and needs fewer features to be tuned. It helps to lower the loss value.

12) LOSS=CATEGORICAL_CROSSENTROPY

Used as the loss function for classification, the output label assigned one hot encoding value in the form of 1s and 0s.

13) EARLY STOPPING

Early stopping will be used for better fitting of training in each iteration while training to keep the model regularized to avoid overfitting with the iterative method.

14) AVERAGING TEST FUNCTION FOR ALL FITTED NEURAL NETWORK

To achieve the confidence of expected model performance, an averaging test function is designed, in which proposed and fitted DNN (5), DNN (3), CNN, and RNNLSTM are used at once for testing predictions. By using SrandadrdScaler() and fitting it against the test data of these models, testing is to be performed.

F. MACHINE LEARNING MODEL CLASSIFICATION

Regarding probabilities, AWID classes with each other, Binomial and Multinomial Logistic Regression, have been implemented with FS of 13 and 76. Logistic Regression is the machine learning-based supervised classification technique that provides discrete outcomes. Binomial is standard LG for binary classes probability prediction. Multinomial is the extension of LG to support multiclass probability prediction by using One-Vs-Rest and One-Vs-One extensions heuristic methods. The LG configured for multinomial by setting up multi_class='ovr' and multi_class='auto'. To implement multinomial problems into multiple binomial classifications problem by using the following:

$$(NumClasses * (NumClasses\check{}1))/2 \tag{3}$$

These classification problems are evaluated by training and testing accuracy, classification report, and confusion matrix.

G. SUMMARY OF PROPOSED MODEL

The proposed model for intrusion detection in network-based IDS specifically targets Wi-Fi networks. It involves receiving and analyzing Wi-Fi frames from network sensors to detect intrusion. The model's implementation employs both machine learning and deep learning techniques. However, this research focuses on a deep-learning approach utilizing CNNs. The model is designed to handle both binary and multiclass classification tasks, relying on a feature set of 76 distinct characteristics as shown in Table 6. It aims to detect intrusions in network environments, particularly those using Wi-Fi technology. The CNN structure for an IDS in WSN includes a convolutional layer for feature extraction, max pooling, and dropout layers to reduce dimensionality and prevent overfitting. The network includes two dense layers for decision-making based on extracted features and additional dropout layers to avoid overfitting. The model's complexity is highlighted by its total trainable parameters of 154,819.

TABLE 6. Summary of proposed model.

Layer(type)	Output Shape	Parameter #
conv1d_1 (Conv1D)	(None, 76, 32)	128
max_pooling1d_1 (MaxPooling1)	(None, 19, 32)	0
dropout_4 (Dropout)	(None, 19, 32)	0
conv1d_2 (Conv1D)	(None, 19, 64)	6208
max_pooling1d_2 (MaxPooling1)	(None, 9, 64)	0
dropout_5 (Dropout)	(None, 9, 64)	0
flatten_1 (Flatten)	(None, 576)	0
dense_5 (Dense)	(None, 256)	147712
dropout_6 (Dropout)	(None, 256)	0
dense_6 (Dense)	(None, 3)	771

Total params: 154,819
Trainable params: 154,819
Non-trainable params: 0

IV. RESULTS AND DISCUSSIONS

In this research, evaluation parameters confusion matrix, accuracy, precision, recall, f-1, micro average, macro average, and support using classification report are calculated for proposed model performance.

A. EXPERIMENTAL RESULTS

All experiments are done to check the performance of the proposed model. Model classification is distributed into binary classification and multiclass classification. Feature sets of 76 and 13 are being considered for experiments. AWID has four types of classes; for binary classification, ''impersonation'', ''infection'', and ''flooding'' classes count in the ''attack'' class and the fourth one is the ''normal'' class. Multiclass classifications ''normal'', ''impersonation'', ''infection'', and ''flooding'' are classified separately.

1) BINARY CLASSIFICATION RESULT BASED ON 76 FEATURES

Table 7 and 8, shows a summary of applied deep learning and machine learning models using 76 features for binary classification with a predictive analysis report. Table 7 evaluates deep learning models like DNNs, CNNs, RNNs, and LR for binary classification tasks. Key performance indicators include Loss, Accuracy, F1 Score, Support, and Micro and Macro Averages. These metrics provide insights into the models' precision, error rates, and effectiveness in identifying threats. The inclusion of Micro and Macro Averages offers a holistic view of model performance across

TABLE 7. Summary of binary classification based on 76 features.

Algorithm	Loss	Accuracy	F1	Support	Micro Average	Macro Average
DNN (5)	0.41	0.95	Att. 0.76 Nor. 0.97	Att. 35288 Nor. 277960	0.946	0.865
DNN (3)	0.47	0.97	Att. 0.88 Nor. 0.98	Att. 35288 Nor. 277960	0.69	0.929
CNN	0.14	0.96	Att. 0.85 Nor. 0.98	Att. 35288 Nor. 277960	0.965	0.915
RNN	0.68	0.93	Att. 0.64 Nor. 0.96	Att. 35288 Nor. 277960	0.923	0.8
LR	_	0.98	Att. 0.883 Nor. 0.987	Att. 35288 Nor. 277960	0.975	0.935
LR (c=10.0)	_	0.94	Att. 0.647 Nor. 0.967	Att. 35288 Nor. 277960	0.93	0.807
MV (LR)	_	0.99	Att. 0.996 Nor. 0.969	Att. 35288 Nor. 277960	0.969	0.969

TABLE 8. Predictive analysis report of binary classification based on 76 features.

MODELS	DETECTION RATE	FALSE ALARM RATE
DNN (5)	272370	10018
DNN (3)	270997	1896
CNN	265735	100
RNN-LSTM	272309	16196
LR	773317	40
LR (c=10)	774247.	28
MV (LR)	773308	27

classes, emphasizing the balance between identifying true positives and minimizing false negatives. In contrast, Table 8 shifts the focus toward operational metrics critical for the practical deployment of IDS within WSNs, namely the Detection Rate and False Alarm Rate. These metrics are indispensable for gauging the models' operational viability, with a high Detection Rate indicating a model's adeptness at accurately identifying threats and a low False Alarm Rate reflecting the model's precision in distinguishing between legitimate and malicious activities. The careful balance between DR and FAR highlighted in this table is crucial for ensuring the IDS's reliability, minimizing the disruption caused by false alarms, and maintaining the integrity of network operations.

2) BINARY CLASSIFICATION RESULT BASED 13 FEATURES

The study focuses on improving security in WSNs through IDS. The researchers evaluated various computational models, including DNNs, CNNs, and (RNNs, to determine their effectiveness and efficiency. Table 9 and 10 , shows a summary of applied deep learning and machine learning models using 13 features for binary classification with a predictive analysis report. Table 9 showcases evaluation metrics for each model, including precision, recall, and F1 score, which help understand each model's accuracy in identifying threats and minimizing false alarms. Table 10 assesses the models from a resource utilization standpoint, including processing times and memory requirements, to determine their feasibility in real-world WSN settings. This analysis is crucial due to limited resources in WSN infrastructures. The goal is to identify models that balance high detection capabilities with efficient resource use, ensuring the proposed IDS solutions are robust in security performance and feasible for practical implementation in WSNs.

3) SUMMARY OF MULTICLASS CLASSIFICATION BASED ON 76 FEATURES

Table 11 and 12, shows the summary of applied deep learning and machine learning models using 76 features for multiclass classification with a predictive analysis report. Table 11 compares various models using advanced metrics like AUC and PR, highlighting models that excel in sensitivity and specificity for intrusion detection tasks. Table 12 evaluates the operational viability of these models in real-world WSN settings, considering scalability, adaptability to network dynamics, and computational resources needed for real-time monitoring and threat mitigation. It guides the selection of robust, practical, and resource-efficient IDS solutions. These tables quantify model performances through traditional metrics and evaluate their readiness and efficiency in real-world WSN applications. This dual perspective ensures the findings are grounded in academic rigor and practical relevance, paving the way for deploying IDS solutions that can navigate the complexities of cybersecurity in WSNs.

4) MULTICLASS CLASSIFICATION RESULTS BASED ON 13 FEATURES

Table 13 and 14, shows the summary of applied deep learning and machine learning models using 13 features for multiclass classification with a predictive analysis report. Table 13 compares IDS models based on their detection capabilities against various cyber threat scenarios, highlighting their effectiveness against sophisticated attack vectors and resilience in adapting to evolving threat landscapes. This comparison helps identify each model's strengths and limitations, providing a nuanced understanding of their operational efficacy in real-world settings. Table 14 focuses on the practical deployment considerations of these IDS models within a WSN environment, evaluating deployment

TABLE 9. Summary of binary classification based 13 features.

Algorithm	Loss	Accuracy	F1	Support	Micro Average	Macro Average
DNN (5)	0.73	0.94	Att. 0.64 Nor. 0.97	Att. 35288 Nor. 277960	0.932	0.805
DNN (3)	0.26	0.94	Att. 0.64 Nor. 0.97	Att. 35288 Nor. 277960	0.932	0.805
CNN	0.14	0.97	Att. 0.88 Nor. 0.98	Att. 35288 Nor. 277960	0.968	0.929
RNN	0.47	0.94	Att. 0.64 Nor. 0.97	Att. 35288 Nor. 277960	0.932	0.805
LR	_	0.918	Att. 0.431 Nor. 0.956	Att. 35288 Nor 277960	0.897	0.64
LR (c=10.0)	_	0.89	Att. 0.000 Nor. 0.940	Att. 35288 Nor. 277960	0.834	0.47
MV (LR)	_	0.996	Att. 0.431 Nor. 0.956	Att. 35288 Nor. 277960	0.897	0.694

TABLE 10. Predictive analysis report of binary classification based 13 features.

MODELS	DETECTION RATE	FALSE ALARM RATE
DNN (5)	277918	18567
DNN (3)	277895	18567
CNN	268142	15
RNN-LSTM	277937	18615
LR	772624	179
LR (c=10)	773254	242
MV (LR)	772614	175

complexities, maintenance requirements, and the impact on network performance and efficiency. It presents data on computational overhead, energy consumption, and integration ease with existing WSN infrastructures. The study aims to bridge the gap between theoretical model performance and practical application feasibility, ensuring the proposed IDS frameworks are technically sound and sustainable options for safeguarding WSNs against various cyber threats, contributing to advancing cybersecurity measures in wireless communications.

B. DISCUSSION OF RESULTS

The IDS for WSNs was developed using DNNs, CNNs, and RNN-LSTMs due to their proven pattern recognition, feature extraction, and sequence analysis capabilities. These models are adept at handling the complexities of WSN data, facilitating the identification of nuanced threat patterns. DNNs excel at extracting hierarchical data representations, making them ideal for discerning complex intrusions. CNNs are celebrated for their efficiency in feature extraction and recognition within high-dimensional datasets, enabling the detection of intricate attack signatures dispersed across the network. RNN-LSTMs stand out for their exceptional ability to analyze temporal data, a critical feature for identifying attacks that unfold over time. Alternative models like SVM and Random Forests were considered but were ultimately not chosen due to their limitations in handling the dataset's complexity and the dynamic nature of WSN traffic. The decision-making process was based on Occam's Razor, advocating for the simplest yet effective solution to the problem. Preprocessing steps, such as null value handling and feature scaling, were implemented to optimize input for the models. Evaluation metrics like detection rate, false alarm rate, accuracy, and loss were chosen to assess IDS performance, ensuring accurate threat identification while minimizing false positives. This methodology underscores the company's commitment to advancing IDS solutions in the WSN domain.

However, the study's reliance on the processed AWID dataset, with significant feature reduction for computational efficiency, introduces potential external validity concerns. This preprocessing may limit the model's generalizability across real-world WSN scenarios, potentially affecting its applicability in diverse operational environments. The high-performance metrics (accuracy, detection rate, and low false alarm rates) highlight the models' effectiveness, but their generalizability to other datasets or real-world scenarios remains uncertain. To address these threats, the research should include a broader range of datasets, incorporating more nuanced evaluation metrics and exploring adaptive feature selection techniques and continuous learning models to improve the detection of novel and evolving cyber threats. This approach addresses the identified validity concerns and strengthens the foundation for deploying effective IDS solutions in diverse and challenging operational settings.

Multiple experiments are conducted using four proposed deep learning models and four machine learning models that are trained, validated, and tested on the AWID dataset, having four classes: ' impersonation'', ''flooding'', ''injection'', and ''normal''. All experiments were performed on 76 preprocessed features and 13 out of 154 for binary and multiclass classification. For binary classification, the best loss value obtained is 0.14 by CNN with FS of 76 and 13, the accuracy value of 0.97 by CNN, and DNN (3) with FS of 76. For multiclass classification, the best loss value obtained is 0.62 by RNN-LSTM with FS of 76 while the accuracy value of 0.92 by DNN (5) and RNN-LSTM with FS 0f 13 as well as by CNN, DNN (5), and DNN (3) with FS of 76.

Moreover, ML-based conducted all experiments having test accuracy of 88.73% while the best training accuracy for binary classification is achieved by binominal of 0.98 having FS of 76, and for multiclass classification multinomial

TABLE 11. Summary of multiclass classification based on 76 features.

Algorithm	Loss	Accuracy	F1	Support	Micro Average	Macro Average
DNN (5)	0.91	0.92	Imp. 0.00 Inj. 0.57 Nor. 0.99	Imp. 18606 Inj. 16682 Nor. 277960	0.908	0.52
DNN (3)	0.79	0.89	Imp. 0.00 Inj. 0.54 Nor. 0.97	Imp. 18606 Inj. 16682 Nor. 277960	0.909	0.52
CNN	1.07	0.91	Imp. 0.00 Inj. 0.57 Nor. 0.98	Imp. 18606 Inj. 16682 Nor. 277960	0.908	0.517
RNN	1.16	0.92	Imp. 0.75 Inj. 0.01 Nor. 0.96	Imp. 18606 Inj. 16682 Nor. 277960	0.868	0.41
LR(OVR)	_	0.89	Imp. 0.00 Inj. 0.153 Nor. 0.95	Imp. 18606 Inj. 16682 Nor. 277960	0.851	0.37
LR(OV0)	_	0.88	Imp. 0.00 Inj. 0.024 Nor. 0.94	Imp. 18606 Inj. 16682 Nor. 277960	0.835	0.32
MV (LR)	_	0.999	Imp. 0.00 Inj. 0.154 Nor. 0.951	Imp. 18606 Inj. 16682 Nor. 277960	0.852	0.3699

TABLE 12. Predictive analysis report of multiclass classification based on 76 features.

MODELS	DETECTION RATE	FALSE ALARM RATE IMPERSONATION	FALSE ALARM RATE INJECTION
DNN (5)	270866	3	4
DNN (3)	272546	690	407
CNN	270724	103	44
RNN-LSTM	270990	1596	16515
LR (OVR)	775140	11	20
LR (OVO)	775140	11	20
MV (LR)	775129	9	20

TABLE 13. Multiclass classification results based on 13 features.

Algorithm	Loss	Accuracy	F1	Support	Micro Average	Macro Average
DNN (5)	2.95	0.919	Imp. 0.00 Inj. 0.57 Nor. 0.99	Imp. 18606 Inj. 16682 Nor. 277960	0.909	0.52
DNN (3)	1.61	0.918	Imp. 0.00 Inj. 0.54 Nor. 0.97	Imp. 18606 Inj. 16682 Nor. 277960	0.909	0.52
CNN	0.65	0.915	Imp. 0.00 Inj. 0.57 Nor. 0.98	Imp. 18606 Inj. 16682 Nor. 277960	0.921	0.593
RNN	0.49	0.933	Imp. 0.75 Inj. 0.01 Nor. 0.96	Imp. 18606 Inj. 16682 Nor. 277960	0.908	0.517
LR(OVR)	_	0.887	Imp. 0.00 Inj. 0.153 Nor. 0.95	Imp. 18606 Inj. 16682 Nor. 277960	0.834	0.32
LR (OVO)	_	0.887	Imp. 0.00 Inj. 0.024 Nor. 0.94	Imp. 18606 Inj. 16682 Nor. 277960	0.834	0.32
MV (LR)	_	0.999	Imp. 0.00 Inj. 0.154 Nor. 0.951	Imp. 18606 Inj. 16682 Nor. 277960	0.834	0.315

TABLE 14. Predictive analysis report of multiclass classification based on 13 features.

MODELS	DETECTION RATE	FALSE ALARM RATE IMPERSONATION	FALSE ALARM RATE INJECTION
DNN (5)	277924	18563	5785
DNN (3)	277168	18560	9863
CNN	268468	37	12
RNN-LSTM	271068	34	579
LR (OVR)	774817	31	134
LR (OVO)	774817	31	134
MV (LR)	774382	8	29

TABLE 15. Performance evaluation of deep learning & machine learning models.

Models	Loss using 13 features		Loss using 76 features		Accuracy using 13 features		Accuracy using 76 features		Micro Average using 13 features		Micro Average using 76 features	
	M	B	M	B	M	B	M	B	M	B	M	B
DNN (5)	1.25	0.73	1.11	0.41	0.92	0.94	0.92	0.95	0.89	0.93	0.91	0.95
DNN (3)	5.79	0.26	0.97	0.47	0.91	0.94	0.92	0.97	0.87	0.93	0.91	0.97
CNN	0.69	0.14	0.92	0.14	0.91	0.97	0.92	0.96	0.89	0.97	0.92	0.97
RNN-LSTM	0.62	0.47	0.88	0.68	0.92	0.94	0.88	0.93	0.91	0.93	0.87	0.92
LR	-	-	-	-	0.89	0.92	0.88	0.98	0.83	0.9	0.86	0.98
LR	-	-	-	-	0.89	0.89	0.89	0.94	0.83	0.83	0.84	0.93
MV (LR)	-	-	-	-	0.99	0.99	0.99	0.99	0.85	0.9	0.85	0.97

TABLE 16. Comparative analysis of existing works using AWID.

Ref.	Method	Feature Set	Binary Accuracy	Multiclass Accuracy	
		Machine Learning Models			
[29]	Linear Support Vector Machine	5	98.22	_	
[17]	RF	15	_	100	
	XGBoost			99.8	
	LGBM			99.99	
	CatBoost			99.98	
[30]	Bagging	34	_	1st stage	2nd stage
	RF			99.41	99.99
	ET			99.57	99.99
	XGBoost			99.55	99.99
	NB			99.49	99.99
				87.85	100
[22]	ANN	20	**99.95**		
[31]	Self-adaptive grasshopper algorithm	8	_	99.15	
[18]	Decision Tree	76	97, 99, 98	_	
	Random Forest	13	99, 99, 98		
	SVM	8	99, 99, 98		
[32]	Cascaded SVM	154	_	1st classi.	2nd class.
				98.75	98.56
[33]	Supervised Clustering Based Classifier	25	_	89.8	
	KNN			88.7	
	SVM			89.1	
[34]	Random Forest	37	99.1	_	
	Bagging		98.97		
	Extra Tree		99.02		
	XGBoost		98.94		
	Proposed Binomial, Multinomial	13, 76	**99.9**	**99.9**	
		Deep Learning Models			
[37]	Unsupervised k-mean clustering	50	94.81	_	
[39]	Deep k-mean clustering	154	94.81	_	
[40]	SAE	71	_	Normal 98.46	
	DNN(7 hidden layers)			Injection 99.99	
	DNN(3 hidden layers)			Impersonation 98.40	
				Flooding 73.12	
[41]	DQN	154	_	95.41	
	DDQN			95.7	
	PG			92.21	
	AC			92.21	
[35]	FFDNN(3 hidden layers)		Min val AC	Min val AC	Min Tst AC
			98.62	98.47	98.59
		154	99.67	99.78	99.77
		26			
[5]	DNN	23	_	95.72	
	Proposed DNN (5)	13	94	92	
		76	95	92	
	Proposed DNN (3)				
		13	94	91	
	Proposed CNN	76	**97**	92	
	Proposed RNN-LSTM	13	**97**	91	
		76	96	92	
		13	94	92	
		76	93	88	

achieved an accuracy of 0.89 having FS of 13. The improved results of LR were obtained by MV with an accuracy of 0.99 for binary and multiclass classification. Detailed comparison of all DL and ML-based models'' experiments is presented in Table 15.

C. COMPARATIVE ANALYSIS

A comparative analysis of the proposed model is shown in Table 16. by which it can be concluded Network Intrusion Detection System for Wi-Fi-based WSN can be implemented using [29] Linear Support Vector Machine, [17] RF, XGBoost, [30], [34] LGBM, CatBoost, Bagging, ET, NB, ANN, FFDNN and with many other algorithms. From all these existing works, it can be concluded that to check the probabilities of AWID classes, LR comes with 99% accuracy, and for ID of WI-FI-based WSNs, CNN has an accuracy of 97% with minimum loss values and the highest micro average.

V. CONCLUSION

The IDS for WSNs uses deep neural networks (DNNs, CNNs, and RNN-LSTMs) to handle WSN data complexities and identify nuanced threat patterns. DNNs excel in hierarchical data representations, CNNs are efficient in feature extraction and recognition, and RNN-LSTMs excel in temporal data analysis. Alternative models like SVM and Random Forests were considered but not chosen due to limitations. The decision-making process is based on Occam's Razor, but the study's reliance. This research proposes a Network Intrusion detection system by designing deep learning and machine learning models in combination with Wi-Fi-based WSN to detect attacks. A reduced group of AWID dataset has 155 attributes with 154 features and 1 class attribute. AWID-CLS-R group contains 4 classes: Normal, Flooding, Injection, and Impersonation. After performing preprocessing, 154 features were reduced to 76, and after applying feature scaling, it turned into 13. All model experiments have been implemented using FS of 13 and 76 to predict binary and multiclass classification. The Deep Learning approach accuracy of 88% to 97% range and loss of 0.62 to 5.79 range are achieved with CNN, RNN-LSTM, DNN (3), and DNN (5), then tested by averaging technique. With the Machine Learning approach, an accuracy of 88% to 98% range is obtained. For binary classification MV, DNN (3), and CNN, while for multiclass classification MV, CNN, and DNN (5) performed well.

VI. FUTURE WORK

As all proposed deep learning models CNN, DNN (5), DNN (3), and RNN-LSTM are fitted with different input shapes and cannot ensemble while using majority voting we can make it work to ensemble them. And averaging test function being designed can only test on a single test set array, and we can update it to work for multiple array testing. Lastly, any other machine learning approach can be tried out for class probabilities with each other.

ACKNOWLEDGMENT

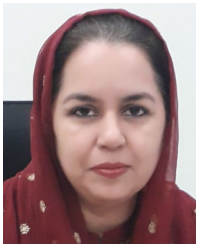
The author would like to thank Prince Sultan University, Riyadh, Saudi Arabia, for their support.

REFERENCES

[1] M. Mittal, R. P. de Prado, Y. Kawai, S. Nakajima, and J. E. Muñoz-Expósito, "Machine learning techniques for energy efficiency and anomaly detection in hybrid wireless sensor networks," *Energies*, vol. 14, no. 11, p. 3125, May 2021.
[2] K. Haseeb, N. Islam, T. Saba, A. Rehman, and Z. Mehmood, "LSDAR: A light-weight structure based data aggregation routing protocol with secure Internet of Things integrated next-generation sensor networks," *Sustain. Cities Soc.*, vol. 54, Mar. 2020, Art. no. 101995.
[3] R. Guetari, H. Ayari, and H. Sakly, "Computer-aided diagnosis systems: A comparative study of classical machine learning versus deep learning-based approaches," *Knowl. Inf. Syst.*, vol. 65, no. 10, pp. 3881–3921, Oct. 2023.
[4] R. Ramadan and K. Medhat, "Intrusion detection based learning in wireless sensor networks," *PLOMS AI*, vol. 2, no. 1, pp. 1–20, 2022.
[5] 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.
[6] S. Mujeeb, T. A. Alghamdi, S. Ullah, A. Fatima, N. Javaid, and T. Saba, "Exploiting deep learning for wind power forecasting based on big data analytics," *Appl. Sci.*, vol. 9, no. 20, p. 4417, Oct. 2019.
[7] 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.
[8] A. Wajahat, J. He, N. Zhu, T. Mahmood, A. Nazir, F. Ullah, S. Qureshi, and S. Dev, "Securing Android IoT devices with GuardDroid transparent and lightweight malware detection," *Ain Shams Eng. J.*, vol. 15, no. 5, May 2024, Art. no. 102642.
[9] E. K. Boahen, S. A. Frimpong, M. M. Ujakpa, R. N. A. Sosu, O. Larbi-Siaw, E. Owusu, J. K. Appati, and E. Acheampong, "A deep multi-architectural approach for online social network intrusion detection system," in *Proc. IEEE World Conf. Appl. Intell. Comput. (AIC)*, Jul. 2022, pp. 919–924.
[10] T. Mahmood, J. Li, T. Saba, A. Rehman, and S. Ali, "Energy optimized data fusion approach for scalable wireless sensor network using deep learning-based scheme," *J. Netw. Comput. Appl.*, vol. 224, Apr. 2024, Art. no. 103841.
[11] Z. Shaukat, A. A. Zulfiqar, C. Xiao, M. Azeem, and T. Mahmood, "Sentiment analysis on IMDB using lexicon and neural networks," *Social Netw. Appl. Sci.*, vol. 2, no. 2, pp. 1–10, Feb. 2020.
[12] P. Satam and S. Hariri, "WIDS: An anomaly based intrusion detection system for Wi-Fi (IEEE 802.11) protocol," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 1, pp. 1077–1091, Mar. 2021.
[13] T. Mahmood, J. Li, Y. Pei, F. Akhtar, S. A. Butt, A. Ditta, and S. Qureshi, "An intelligent fault detection approach based on reinforcement learning system in wireless sensor network," *J. Supercomput.*, vol. 78, no. 3, pp. 3646–3675, Feb. 2022.
[14] 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.
[15] L. Tao and M. Xueqiang, "Hybrid strategy improved sparrow search algorithm in the field of intrusion detection," *IEEE Access*, vol. 11, pp. 32134–32151, 2023.
[16] A. Singh, J. Amutha, J. Nagar, and S. Sharma, "A deep learning approach to predict the number of k-barriers for intrusion detection over a circular region using wireless sensor networks," *Expert Syst. Appl.*, vol. 211, 2023, Art. no. 118588.
[17] S. Bhandari, A. K. Kukreja, A. Lazar, A. Sim, and K. Wu, "Feature selection improves tree-based classification for wireless intrusion detection," in *Proc. 3rd Int. Workshop Syst. Netw. Telemetry Analytics*, Jun. 2020, pp. 19–26.
[18] 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.
[19] S. Rajasoundaran, S. V. N. S. Kumar, M. Selvi, K. Thangaramya, and K. Arputharaj, "Secure and optimized intrusion detection scheme using LSTM-MAC principles for underwater wireless sensor networks," *Wireless Netw.*, vol. 30, no. 1, pp. 209–231, 2024.

[20] I. A. Qureshi, K. A. Bhatti, J. Li, T. Mahmood, M. I. Babar, and M. M. Qureshi, "GFCO: A genetic fuzzy-logic channel optimization approach for LR-WPAN," *IEEE Access*, vol. 11, pp. 88219–88233, 2023.
[21] U. S. K. P. M. Thanthrige, J. Samarabandu, and X. Wang, "Machine learning techniques for intrusion detection on public dataset," in *Proc. IEEE Can. Conf. Electr. Comput. Eng. (CCECE)*, May 2016, pp. 1–4.
[22] M. A. Rahman, A. T. Asyhari, O. W. Wen, H. Ajra, Y. Ahmed, and F. Anwar, "Effective combining of feature selection techniques for machine learning-enabled IoT intrusion detection," *Multimedia Tools Appl.*, vol. 80, no. 20, pp. 31381–31399, Aug. 2021, doi: 10.1007/s11042-021-10567-y.
[23] J. He, N. Zhu, T. Mahmood, A. Nazir, S. Qureshi, F. Ullah, and M. S. Pathan, "Outsmarting Android malware withcutting-edge feature engineering andmachine learning techniques," Tech. Rep., 2023.
[24] S. Gavel, A. S. Raghuvanshi, and S. Tiwari, "An optimized maximum correlation based feature reduction scheme for intrusion detection in data networks," *Wireless Netw.*, vol. 28, no. 6, pp. 2609–2624, Aug. 2022.
[25] S. B. Park, H. J. Jo, and D. H. Lee, "G-IDCS: Graph-based intrusion detection and classification system for CAN protocol," *IEEE Access*, vol. 11, pp. 39213–39227, 2023.
[26] I. A. Kandhro, S. M. Alanazi, F. Ali, A. Kehar, K. Fatima, M. Uddin, and S. Karuppayah, "Detection of real-time malicious intrusions and attacks in IoT empowered cybersecurity infrastructures," *IEEE Access*, vol. 11, pp. 9136–9148, 2023.
[27] E. K. Boahen, B. E. Bouya-Moko, F. Qamar, and C. Wang, "A deep learning approach to online social network account compromisation," *IEEE Trans. Computat. Social Syst.*, vol. 10, no. 6, pp. 3204–3216, Dec. 2023.
[28] H. Shirazi, S. R. Muramudalige, I. Ray, A. P. Jayasumana, and H. Wang, "Adversarial autoencoder data synthesis for enhancing machine learning-based phishing detection algorithms," *IEEE Trans. Services Comput.*, pp. 1–13, 2023.
[29] S. J. Lee, P. D. Yoo, A. T. Asyhari, Y. Jhi, L. Chermak, C. Y. Yeun, and K. Taha, "IMPACT: Impersonation attack detection via edge computing using deep autoencoder and feature abstraction," *IEEE Access*, vol. 8, pp. 65520–65529, 2020.
[30] A. A. Reyes, F. D. Vaca, G. A. C. Aguayo, Q. Niyaz, and V. Devabhaktuni, "A machine learning based two-stage Wi-Fi network intrusion detection system," *Electronics*, vol. 9, no. 10, p. 1689, Oct. 2020.
[31] A. K. Shukla, "Detection of anomaly intrusion utilizing self-adaptive grasshopper optimization algorithm," *Neural Comput. Appl.*, vol. 33, no. 13, pp. 7541–7561, Jul. 2021.
[32] Y. Jian, L. Jian, and X. Dong, "Research on network intrusion detection based on improved machine learning method," *Int. J. Netw. Secur.*, vol. 24, no. 3, pp. 533–540, 2022.
[33] G. Granato, A. Martino, L. Baldini, and A. Rizzi, "Intrusion detection in Wi-Fi networks by modular and optimized ensemble of classifiers: An extended analysis," *Social Netw. Comput. Sci.*, vol. 3, no. 4, p. 310, Jul. 2022.
[34] A. Das, "Design and development of an efficient network intrusion detection system using ensemble machine learning techniques for WiFi environments," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 4, pp. 1–12, 2022.
[35] 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.
[36] T. Mahmood, F. Akhtar, K. U. Rehman, M. Azeem, A. I. Mudassir, and S. M. Daudpota, "Introducing robustness in DBR routing protocol," *Int. J. Commun. Netw. Distrib. Syst.*, vol. 24, no. 3, pp. 316–338, 2020.
[37] M. E. Aminanto and K. Kim, "Improving detection of Wi-Fi impersonation by fully unsupervised deep learning," in *Information Security Applications*. Jeju Island, South Korea: Springer, 2018, pp. 212–223.
[38] B. Ali, T. Mahmood, M. Abbas, M. Hussain, H. Ullah, A. Sarker, and A. Khan, "Leach robust routing approach applying machine learning," *Int. J. Comput. Sci. Netw. Secur.*, vol. 19, no. 6, pp. 18–26, 2019.
[39] K. Kim, M. E. Aminanto, H. C. Tanuwidjaja, K. Kim, M. E. Aminanto, and H. C. Tanuwidjaja, "Deep feature learning," in *Network Intrusion Detection using Deep Learning: A Feature Learning Approach*. Berlin, Germany: Springer, 2018, pp. 47–68.
[40] S. Wang, B. Li, M. Yang, and Z. Yan, "Intrusion detection for WiFi network: A deep learning approach," in *Proc. Int. Wireless Internet Conf.* Cham, Switzerland: Springer, 2019, pp. 95–104.
[41] M. Lopez-Martin, B. Carro, and A. Sanchez-Esguevillas, "Application of deep reinforcement learning to intrusion detection for supervised problems," *Exp. Syst. Appl.*, vol. 141, Mar. 2020, Art. no. 112963.
[42] M. E. Aminanto and K. Kim, "Detecting impersonation attack in WiFi networks using deep learning approach," in *Information Security Applications*, Jeju Island, (South) Korea. Springer, 2017, pp. 136–147.
[43] 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.*, pp. 1–13, Mar. 2021.
[44] B. Alenazi and H. E. Idris, "Wireless intrusion and attack detection for 5G networks using deep learning techniques," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 7, pp. 1–6, 2021.
[45] Q. Liu, D. Wang, Y. Jia, S. Luo, and C. Wang, "A multi-task based deep learning approach for intrusion detection," *Knowledge-Based Syst.*, vol. 238, Feb. 2022, Art. no. 107852.
[46] Y. Chen, Q. Lin, W. Wei, J. Ji, K.-C. Wong, and C. A. C. Coello, "Intrusion detection using multi-objective evolutionary convolutional neural network for Internet of Things in fog computing," *Knowl.-Based Syst.*, vol. 244, May 2022, Art. no. 108505.
[47] S. Gavel, A. S. Raghuvanshi, and S. Tiwari, "A novel density estimation based intrusion detection technique with Pearson's divergence for wireless sensor networks," *ISA Trans.*, vol. 111, pp. 180–191, May 2021.
[48] D. B. Basnet and H. A. Kholidy, "An empirical Wi-Fi intrusion detection system: A master's project presented to the department of network and computer security in partial fulfillment of the requirements for the master of science degree," Dept. Netw. Comput. Secur., College Eng., SUNY Polytech. Inst., 2020.

HALIMA SADIA received the B.S. degree in computer science from the University of Engineering and Technology, Lahore, Pakistan, in 2020, and the M.S. degree in computer science from Lahore College for Women University, Lahore, in 2022. She is currently a Visiting Lecturer with the Department of Computer Science, Government College Women University, Sialkot, Pakistan. Her research interests include machine learning, computer networks, and data science.



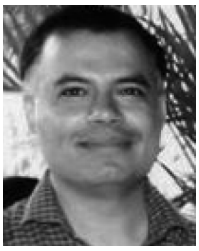
SAIMA FARHAN is currently the Chairperson and an Associate Professor with the Department of Computer Science, Lahore College for Women University, Lahore, Pakistan. She has authored a number of books and research articles in ISI and Scopus-indexed journals. Her research interests include image processing, medical image analysis, and machine learning. She has been a part of various research seminars, conferences, paper presentations, and research article reviews. She has demonstrated vast success in acquiring the latest trends in technology and adopting new pedagogies.

YASIN UL HAQ received the B.S. degree in software engineering from the University of Sargodha, Pakistan, in 2013, and the M.Sc. and Ph.D. degrees in computer science from the University of Engineering and Technology, Lahore, Narowal Campus, Pakistan, in 2017 and 2023, respectively. He is currently a Lecturer with the Department of Computer Science and Engineering, University of Engineering and Technology, Lahore. His research interests include machine learning, remote sensing, web semantics, software engineering, and big data.

RABIA SANA received the B.S. degree in software engineering from the University of Sargodha, in 2013, and the M.Sc. degree in computer science from the Virtual University of Pakistan, in 2020. She is currently a Lecturer of computer science with the University of Engineering and Technology, Lahore, Narowal Campus. Her research interests include software engineering, semantic web, SQA, information systems, machine learning, and database systems.

TARIQ MAHMOOD received the master's degree in computer science from the University of Lahore, Pakistan, and the Ph.D. degree in software engineering from Beijing University of Technology, China. He is currently an Assistant Professor with the Faculty of Information Sciences, University of Education, Vehari Campus, Vehari, Pakistan. He is a renowned expert in image processing, healthcare informatics and social media analysis, ad-hoc networks, and WSN. He has contributed more than 25 research articles in well-reputed international journals and conferences. His research interests include image processing, social media analysis, medical image diagnosis, machine learning, and data mining. He aims to contribute to interdisciplinary research of computer science and human-related disciplines. He is an Editorial Member and a Reviewer of various journals, including *PloS One*, *The Journal of Supercomputer*, *Journal of digital Imaging*, and *International Journal of Sensors, Wireless Communications and Control*.

SAEED ALI OMER BAHAJ received the Ph.D. degree from Pune University, India, in 2006. He is currently an Associate Professor with the Department of Management Information Systems, College of Business Administration, Al-Kharj. He is also an Associate Professor with the Computer Engineering Department, Hadramout University, Yemen, and the MIS Department, College of Business Administration (COBA), Prince Sattam Bin Abdulaziz University. His main research interests include artificial intelligence, information management, forecasting, information engineering, big data, and information security.



AMJAD REHMAN KHAN (Senior Member, IEEE) received the Ph.D. and Postdoctoral degrees (Hons.) from the Faculty of Computing, Universiti Teknologi Malaysia, in 2010 and 2011, respectively, with a specialization in forensic documents analysis and security. He is currently a Senior Researcher with the Artificial Intelligence and Data Analytics Laboratory, College of Computer and Information Sciences (CCIS), Prince Sultan University, Riyadh, Saudi Arabia. He is the author of more than 200 ISI journal articles and conferences. He is currently a PI in several funded projects and also completed projects funded by MOHE Malaysia and Saudi Arabia. His research interests include data mining, health informatics, and pattern recognition. He received the Rector Award for the 2010 Best Student from Universiti Teknologi Malaysia.