Journal of Big Data

RESEARCH Open Access

Cyberattack detection in wireless sensor networks using a hybrid feature reduction technique with AI and machine learning methods

Mohamed H. Behiry[1,2*] and Mohammed Aly[1]

*Correspondence: m.behiry@science.menofia.edu.eg

[1] Department of Artificial Intelligence, Faculty of Artificial Intelligence, Egyptian Russian University, Badr City 11829, Egypt
[2] Department of Computer Science, Faculty of Science, Menofia University, Shibin El Kom 32611, Egypt

Abstract

This paper proposes an intelligent hybrid model that leverages machine learning and artificial intelligence to enhance the security of Wireless Sensor Networks (WSNs) by identifying and preventing cyberattacks. The study employs feature reduction techniques, including Singular Value Decomposition (SVD) and Principal Component Analysis (PCA), along with the K-means clustering model enhanced information gain (KMC-IG) for feature extraction. The Synthetic Minority Excessively Technique is introduced for data balancing, followed by intrusion detection systems and network traffic categorization. The research evaluates a deep learning-based feed-forward neural network algorithm's accuracy, precision, recall, and F-measure across three vital datasets: NSL-KDD, UNSW-NB 15, and CICIDS 2017, considering both full and reduced feature sets. Comparative analysis against benchmark machine learning approaches is also conducted. The proposed algorithm demonstrates exceptional performance, achieving high accuracy and reliability in intrusion detection for WSNs. The study outlines the system configuration and parameter settings, contributing to the advancement of WSN security.

Keywords: WSN, Hybrid reduction, ML, AI, NSL-KDD, UNSW-NB 15, CICIDS 2017

Introduction

The usage of artificial intelligence AI for cyberattack detection in wireless sensor networks with a hybrid feature reduction technique involves developing a system that can effectively detect and classify cyberattacks in WSN environments. The system combines both machine learning and deep learning techniques to reduce the high-dimensional feature space while improving intrusion detection performance. This is achieved by utilizing a hybrid feature reduction technique that incorporates K-means clustering and entropy-based mutual information feature ranking to extract and rank the most relevant features. The system is then trained using a feed-forward deep neural network to accurately categorize network traffic. Overall, the aim is to provide early detection and learning systems with high-performance features for efficient cyberattack detection and

prevention in WSN environments. The Wireless Sensor Network is being destroyed by cyberattacks (WSN. We developed WSN employing cyber-security technologies like machine learning in order to recognize and counter risks linked to WSN (ML). For artificial intelligence models, specialized cyber-security defense and protection solutions are needed. Information systems, Computers, networks, servers, and data must be protected of WSN-related threats with integrity, availability, and confidentiality as a minimum. Maintaining cyber security measures to safeguard sensitive information from online thieves. Virtual computers, cloud services, and network topologies are all protected by cybersecurity, which also helps to stop cybercrimes and aids in forensic investigations. Because the DNS server lacks adequate security, it requires outside protection to stop hackers from stealing its data. By implementing cyber security, this may be done to stop unauthorized access by cybercriminals.

The technique of protecting computer and mobile networks, software, servers, and electronic systems against viruses and malware is known as cybersecurity. Over 10 billion more records have been added to the menace of global cybercrime. In the US, NIST developed a framework for cyber security. Machine learning (ML), a subset of artificial intelligence, is used in cyber-security applications including prediction systems and the detection of zero-day attacks. The four types of machine learning (ML) methodologies are reinforcement, semi-supervised, unsupervised, and supervised. ML is designed for supply in consistent circumstances. Cyberattacks might therefore cause an unstable situation. A group of machine learning algorithms that go through several stages and are trained on various datasets may be thought of as deep learning (DL). In light of the growth of cybercrime, cybersecurity is detecting attacks in WSNs to safeguard shared and stored information and data. Many machine learning methods may render simulated attackers useless for SCADA and VANET intrusion detection systems. Concerns the use of machine learning's core and subcategories in cyber-security to identify malware, spam, rejection attacks, and biometric identification. By creating a brand-new dataset, ML methods utilising the MQTT protocol were recommended for categorizing attacks.

The goals of WSN security that we are going to discuss here are data secrecy, data availability, data authenticity and integrity, data freshness, self-organization, time synchronization, and secure localization.

Threats and attacks in WSN: Performer, objectives, and layer-wise features can be used to classify attackers.

I. Attacks having a particular objective, which fall under either the active attack or passive attack categories.
II. Performer-oriented attacks, which fall under either the inside attacks or outside attacks categories.
III. Layer-oriented attacks, which target the data link, physical, transport, or network levels.

Motivated by the goals of WSN security, a deep feed forward neural network (DFFNN) model with k-means clustering (KMC) and information gain (IG) methods is proposed for attack with the main contributions are described below:

1. The data is over-sampled and cleaned using the SMOTE-based ENN method, which also produces balanced data for further processing.
2. Using the optimum features retrieved from the dataset, DLFFNN approach is proposed to evaluate the validity of the models.
3. The KMC-IG approach, created to retrieve the best features from datasets including UNSW-NB15, NSL-KDD, and CICIDS2017.

In this work, three widely used datasets—NSL-KDD, UNSW-NB 15, and CICIDS 2017—are taken into consideration for evaluating the proposed work. For each dataset, the recommended approach's accuracy, precision, recall, and F-measure are evaluated under the full features and reduced features conditions. The outcomes of the proposed DFFNN-KMC-IG are also contrasted with those of benchmark machine learning methodologies. This approach incorporates deep learning and machine learning in three stages, including feature reduction, extraction of features, and categorization. These procedures are required to halt the reduction in resource availability caused by early attack detection.

The structure of this paper is organized as follows. Section "Hyperparameter tuning" focuses on the related work. Section "Preventing overfitting" knowledge and background which consists of four parts as follows: part 1 explains types of Cyber Attacks such as Malware Phishing, Man in the middle of the attack, SQL injection, and DNS tunnelling; part 2 includes few instances of cyberattacks within 2022 as Theft of Crypto.com, Breach of data at the Red Cross, and Cash app data breach. part three discusses significance of Cybersecurity, while fourth part contains the types of Cyber Security such Cloud security, Mobile security, Security with Zero Trust, Network security, Application security, IoT [1, 2], and End-point security. Section "Early stopping" focuses on Research Methodology including Proposed architecture workflow and algorithms which are "Data pre-processing stage" that includes Encoding Features Based on Labels, and Feature Normalization using Logarithmic technique, "Data splitting" stage, "Feature extraction and selection using KMC-IG-based FES", "Data balancing using SMOTE and ENN stage", "Training and validation stage" which explains DFNN and some Traditional machine learning (ML) Models. Section "Experiments and results" presents Experiments and Results which includes Datasets Description and Modelling, Binary Classification and Multi-class Classification with the Full and Reduced Feature Set, and comparisons with current related work. Sect. "Conclusion" is devoted to the conclusion of this study.

Related work

In their work, Kaur Saini et al. [3] conducted an evaluation of cyberattacks, while Chelli [4] investigated security issues and challenges in wireless sensor networks, including attacks and countermeasures. Daojing He et al. [5] focused on the cybersecurity defense of wireless sensor networks for smart grid monitoring. Padmavathi and Shanmugapriya [6] surveyed attacks in wireless sensor networks, covering security mechanisms and challenges. Al-Sakib Khan Pathan et al. [7] investigated security issues and challenges in wireless sensor networks, while Perrig et al. [8] discussed security in wireless sensor networks. Jian-hua Li [9] conducted a survey on the intersection of cybersecurity and artificial intelligence. Handa et al. [10] reviewed machine learning in cybersecurity, and Thomas et al. [11] investigated machine learning approaches for cybersecurity analytics. Gaganjot et al.

[12] discussed secure cyber-physical systems for smart cities, while Boussi and Gupta [13, 14] developed a framework for combating cybercrime. Kumar [15] researched artificial intelligence-based approaches for intrusion detection. Shahnaz Saleem et al. [16] focused on network security threats in wireless body area networks, and Kalpana Sharma [17] outlined security issues in wireless sensor networks. Martins and Guyennet [18] provided a brief overview of wireless sensor network attacks and security procedures, while Anitha S. Sastry [19] examined security threats at every layer of wireless sensor networks. Kaplantzis [20] investigated security approaches for wireless sensor networks, and Chris and Wagner [21] explored secured routing and countermeasures. Yanli Yu et al. [22] investigated trust algorithms in wireless sensor networks, including hazard analysis. Xu et al. [23] explored the feasibility of launching and detecting jamming attacks in wireless networks, while Xu [24] investigated safeguarding wireless sensor networks from interference through channel surfing. Finally, Sohrabi [25] explored protocols for self-organizing wireless sensor networks. David and Scott [26] investigated Denial-of-Service attacks and defense of attacks and making Protections in Wireless Sensor Networks. Consolidated Detection of Node Replication Attacks in Sensor Networks was explored by Parno and Gligor [27]. A review of important management systems in wireless sensor networks was conducted by Xiao et al. [28]. Abhishek Jain et al. [29] investigated Wireless Sensor Network Cryptographic Protocols. Daniel E. Burgner is an American businessman. Luay Wahsheh [30] investigated Wireless Sensor Network Cybersecurity. Zhu et al. [31] investigated effective security solutions for large-scale wireless sensing networks. Culler and Hong [32] conducted research on Wireless Sensor Networks. Makhija et al. [33] used Machine Learning Techniques to classify attacks on MQTT-based IoT systems. Wang [34] explored an ensemble technique based on hybrid spectral segmentation in sensor networks. Zhang [35], on the other hand, used adversarial feature extraction to defend versus evasion assaults. Regarding some related works to the same datasets, we found that Tavallaee et al. [36] studied in details NSL-KDD dataset and the KDD CUP 99 data set. Sonule et al. [37] focused on UNSWNB15 Dataset and ML. Sharafaldin et al. [38] gave the attention toward generating a new intrusion detection dataset especially CICIDS2017 Dataset and intrusion traffic characterization. Aly and Alotaibi studied the modified gedunin using ML [39]. The referenced literature covers a broad spectrum of machine learning applications in security domains. Johri et al. [40] provide an overarching view of machine learning algorithms for intelligent systems, setting the stage for diverse applications. Rikabi and Hazim [41] propose an innovative fusion of encryption and steganography to enhance communication system security. Ahmad et al. [42] offer a comprehensive perspective on challenges in securing wireless sensor networks using machine learning. Ismail et al. [43] conduct a comparative analysis of machine learning models for cyber-attack detection in wireless sensor networks, while Khoei et al. [44] explore dynamic techniques against GPS spoofing attacks on UAVs. Karatas [45] focuses on refining machine learning-based intrusion detection systems, specifically addressing dataset challenges. Together, these studies underscore the vital role of machine learning in fortifying security measures across various technological domains, providing diverse strategies to tackle evolving threats.

In continuation of related works, regarding to traditional approaches to WSN Security.

traditional methods have laid the groundwork for securing Wireless Sensor Networks (WSNs). Cryptographic techniques, as discussed by Dong et al. [46], play a vital role

in ensuring data confidentiality and integrity. Access control mechanisms, as explored by Zhang et al. [47], contribute to regulating network access, preventing unauthorized intrusions. While effective, traditional methods may face challenges in adapting to the dynamic nature of cyber threats.

Machine learning-based intrusion detection in WSNs

Machine learning (ML) techniques have been extensively explored for intrusion detection in WSNs. Recent studies, such as the work by Li et al. [48], utilize decision trees, support vector machines, and ensemble methods to leverage features extracted from network traffic data. Despite their effectiveness, ML-based methods may encounter challenges in adapting to new and evolving attack patterns.

Deep learning in WSN security

Deep learning techniques have gained attention for enhancing WSN security. Research by Wang et al. [49] explores the use of deep neural networks and attention mechanisms to capture intricate patterns in network data. Despite promising results, challenges related to interpretability and the need for substantial labeled data persist in deep learning approaches, as discussed by Chen et al. [50].

Clustering techniques for anomaly detection

Clustering algorithms, particularly K-means clustering, continue to be applied for anomaly detection in WSNs. The study by Kim et al. [51] demonstrates the use of clustering to group similar network behaviors, aiding in anomaly detection by identifying deviations from established norms. While effective, the dynamic nature of WSNs may influence the performance of clustering methods.

Feature reduction methods in WSN security

Feature reduction remains critical for enhancing the efficiency of intrusion detection systems. Recent studies, such as the work by Jingjing et al. [52], explore techniques like Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) for reducing the dimensionality of data. These methods contribute to the identification of key features associated with specific attack categories.

Comparative studies and benchmarking

Comparative studies, such as the one conducted by Zhao et al. [53], benchmark various intrusion detection approaches in WSNs. These studies assess the strengths and weaknesses of different methods in terms of accuracy, precision, recall, and F-measure. Benchmarking provides insights into the relative performance of different techniques, guiding the selection of optimal models for specific WSN scenarios.

Challenges and open issues

Challenges persist in WSN security, as highlighted by recent research. Adapting to dynamic network conditions, ensuring scalability, and addressing the limitations of

existing approaches remain open issues. The trade-off between detection accuracy and resource consumption is a constant challenge, as discussed by Liu et al. [54].

Summary and positioning

In the dynamic landscape of WSN security, recent literature reflects a continuous evolution from traditional methods to sophisticated machine learning and deep learning approaches. The proposed Deep Forward Neural Network (DFNN) Classification Mode, as outlined in our study, seeks to address challenges observed in previous works by integrating feature reduction, clustering, and deep learning for robust intrusion detection and classification in WSNs.

This "Related works" section includes recent references and provides a detailed analysis of existing literature, establishing the context for the proposed DFNN Classification Mode in the rapidly advancing field of WSN security research.

The following topics have not previously been studied, which they represent the research gap in current related works:

- It has not been investigated how to identify cyberattacks in wireless sensor networks using a hybrid feature reduction technique and machine learning.
- DLFFNN methodology is not combined with the SMOTE-based ENN method.
- While K-means Clustering-based Information Gain is utilized instead, the KMC-IG technique is not employed to extract the best features from datasets like UNSW-NB15, NSL-KDD, and CICIDS2017 (KMC-IG).

Knowledge and background

i. Types of cyber attacks

A cruel and unlawful attempt to steal priceless information and data from a specific person without that person's knowledge is known as a cyber-attack. Hackers are profiting off valued firms' sensitive data as cyberattacks rise every year. Cybercrime has cost more than 500,000 dollars over the last few years. The most typical forms of cyberattacks are as follows:

a) *Malware*: The word "malware" is used to refer to unapproved programmes, applications, viruses, and worms. When a consumer hits the email links and message links and downloads unapproved programmes, malware software is installed. The virus can perform the following once it has been installed.

1. Block internal security modules, for one.
2. Introduce dangerous software into the system.
3. Constant data transmission from the computer's hard disc.

b) *Phishing:* Phishing is a generic term for the fraudulent activity of repeatedly sending emails from the same source with personal information in them. This kind is frequently used to get financial information, such as credit card information. The

hacker infects computers and mobile devices with malware through the email link in order to steal crucial data.

c) *Man in the middle of the attack:* The man-in-the-middle assault, commonly referred to as a bug attack, typically involves hackers who generate network traffic. After gaining access to the network, the hacker will implant a flaw in the system that will enable the hacker to access information from all of the victim's machines. When a user authenticates to public WiFi, the hacker exploits weaknesses in the network to generate traffic.
d) *SQL injection:* When hackers insert code into the server that contains a virus or access control code, this is known as a structured query language (SQL) injection assault. The hacker gains access through this gateway when a victim runs the malicious code on their computer, allowing them to steal personal information.
e) *DNS tunnelling:* DNS tunnelling delivers HTTP or another protocol via DNS in order to communicate with network-connected devices that are not linked to the DNS server protocol over a certain port number. Once connected, the hacker can use the DNS protocol to steal information online.

ii. Listed below are a few instances of cyberattacks within 2022.

1) *Theft of Crypto.com*: This assault took place on January 17 and targeted the bitcoin wallets of 500 users. The hacker stole approximately 18 million dollars in bitcoins, 15 million dollars in Ethereum, and other cryptocurrencies.
2) *Breach of data at the Red Cross*: The servers containing the personal data of almost 500,000 people who received assistance from the red-cross movement were attacked by hackers in January. The compromised server contains information about the company as well as the victims' personal and family information.
3) *Cash app data breach*: Cash App acknowledged that a hacker with broad access to the business had gained access to the cash servers. In addition, this breach included hacking of client information, company data, account numbers, inventory data, portfolio values, and other confidential financial data.

iii. Significance of cyber security

Cybersecurity needs to be a top priority for every nation's military, government, commercial, private, medical, and financial organisations since they store a lot of data on servers, the cloud, and other gadgets. Overall, whether the data is sensitive or not, it can still pose issues for the business if intellectual, economic, financial, or any other type of data is open to illegal access or public inspection. There is a personal as well as an organisational future if the security of any application or website is poor. All firms are creating their own protection software to shield their sensitive data from security risks and assaults. Cybersecurity is crucial because it guards against viruses and malware and safeguards information as well as our computer systems. Cybercrimes are on the rise, and businesses and organisations, particularly those in the health, economic, and national safety sectors, need to take extra

precautions to secure their data because the future of any nation depends on it. Every firm need cyber security to safeguard its critical data information from hackers. The nation's top intelligence officials issued a warning in April 2013 that cyberattacks and online surveillance posed a threat to national security concerns. Every person must be concerned about cyber security. We should maintain security while the system or files are connecting to the internet to prevent cybercrimes and decrease the chance of cyber-attacks.

iv. Types of cyber security

Various forms of cybersecurity exist, including Cloud security, Mobile Security, Zero trust, Network security, Application Security, IOT security, End-point security. Here the explanations of them are indicated as:

1) *Cloud security:* Cloud computing is another name for cloud security. Many businesses nowadays are implementing cloud computing for their operations. A primary concern is ensuring cloud security. To safeguard the whole organization's cloud communications and architecture, cloud safety consists of solutions, policies, and services. A third-party solution is frequently provided by cloud security companies to safeguard an organization's cloud data.
2) *Mobile security:* Malicious software, phishing scams, and instant messaging assaults must be prevented even on locked mobile phones, computers, and other tiny electronic devices. These hacks are stopped by mobile security systems, which also protect user data. When connected to the assets of the company, mobile device management (MDM) solutions will provide or guarantee access to the specific application.
3) *Security with zero trust:* Zero-trust architecture is another name for zero-trust security (ZTA). The conventional security model places an emphasis on the perimeter and calls for the construction of fortified walls around the organization's most important assets. However, there are several severe problems with this strategy, including possible risks. A strategic approach to cyber security is zero-trust security, which aims to keep the validity of digital contact.
4) *Network security:* Only in this area do attacks often occur. To stop hackers from hacking networks, there are words and programmes for network security. Data integrity and usability on personal and computer networks will be safeguarded. Among the strategies used to avoid data theft include information loss prevention (DLP), identification access management (IAM), and network access control (NAC), and next-generation firewall restrictions.
5) *Application security:* Application security refers to security at the operating system. Due to their direct internet connection, web apps are vulnerable to data theft. Weaknesses in online applications such cross-site scripting, failed authentication, and injection. Unauthorized contact with apps and APIs is prevented by application security
6) *IoT:* IoT security is a procedure used to protect IoT systems from dangers. The effectiveness of IoT devices boosts productivity in today's environment where the Internet of Things plays a significant role in all facets of the enterprise. Tools for Internet of

Things security aid in defending against dangers and breaches. Device identification, device authentication, and data encryption can all help to safeguard IoT systems.

7) *End-point security:* Remote computer access occurs in every company. Controlling an organization's end or entrance points, such as computers, laptops, and electrical controllers, is known as end-point security.

Research methodology

This study proposes using the K-means clustering model to improve information gain for feature reduction/extraction and ranking (KMC-IG). Additionally, a Synthetic Minority Over-sampling Technique is suggested. The final critical stage involves the classification of network traffic and intrusion protection systems. The network traffic feature datasets undergo several stages in succession, and for each dataset, the accuracy, precision, recall, and F-measure of the proposed approach are evaluated under the full features and reduced features scenarios. Furthermore, the performance of the proposed DFFNN-KMC-IG is compared to that of benchmark machine learning algorithms. By combining the strengths of DL and ML, the proposed hybrid model adapts the reduced attributes to improve their quality.

Wireless Sensor Networks intrusion detection systems (WSN-IDS) are crucial for ensuring the security of networked computer systems, but many WSN-IDS still struggle with efficiency. The feature space grows, the accuracy of existing ML-based WSN-IDS techniques effectively decreases. The feature extraction and optimization are performed using the K-means clustering with information gain approach proposed in this work.

In Fig. 1, we can extract features from packet capture using Network Traffic Data Packet (PCAP). The Pre-processing Step from Network Traffic Features Datasets can then be represented by Feature Representation using Label Encoding, or Feature Normalization using Logarithmic or Min–Max approaches. The Data Splitting Step then included the Training Set, Validation Set, and Testing Set. They all use KMC-IG for Feature Reduction and Selection to produce Training Set Reduced Features, Validation Set Reduced Features, and Testing Set Reduced Features. To Training Set Reduced Features, Data Balancing was implemented using SMOTE and ENN Stage. This implementation resulted in all Training and Validation Stages Developing and Training a suggested Deep Forward Neural Network (DFNN) Classification Model and Some Conventional Machine Learning (ML) Models, and this is the same result from Validation Set Reduced Features without balance. The following stage is the evaluation stage, which involves testing the trained DFNN model as well as other trained ML models. Confusion matrices, accuracy, F1-score, recall, and precision are all included in the classification Report. Lastly, the Comparisons Stage compares the acquired findings to some current relevant outcomes. Here, the proposed architecture workflow as in Fig. 1.

Each of these elements performs a crucial role and significantly affects the effectiveness of the WSN-IDS model. The design of the planned work for developing WSN-IDS is shown in Fig. 1.

Certainly! Let's delve into an overview of how the proposed Deep Forward Neural Network (DFNN) Classification Mode works, including details on the layers used in its architecture.

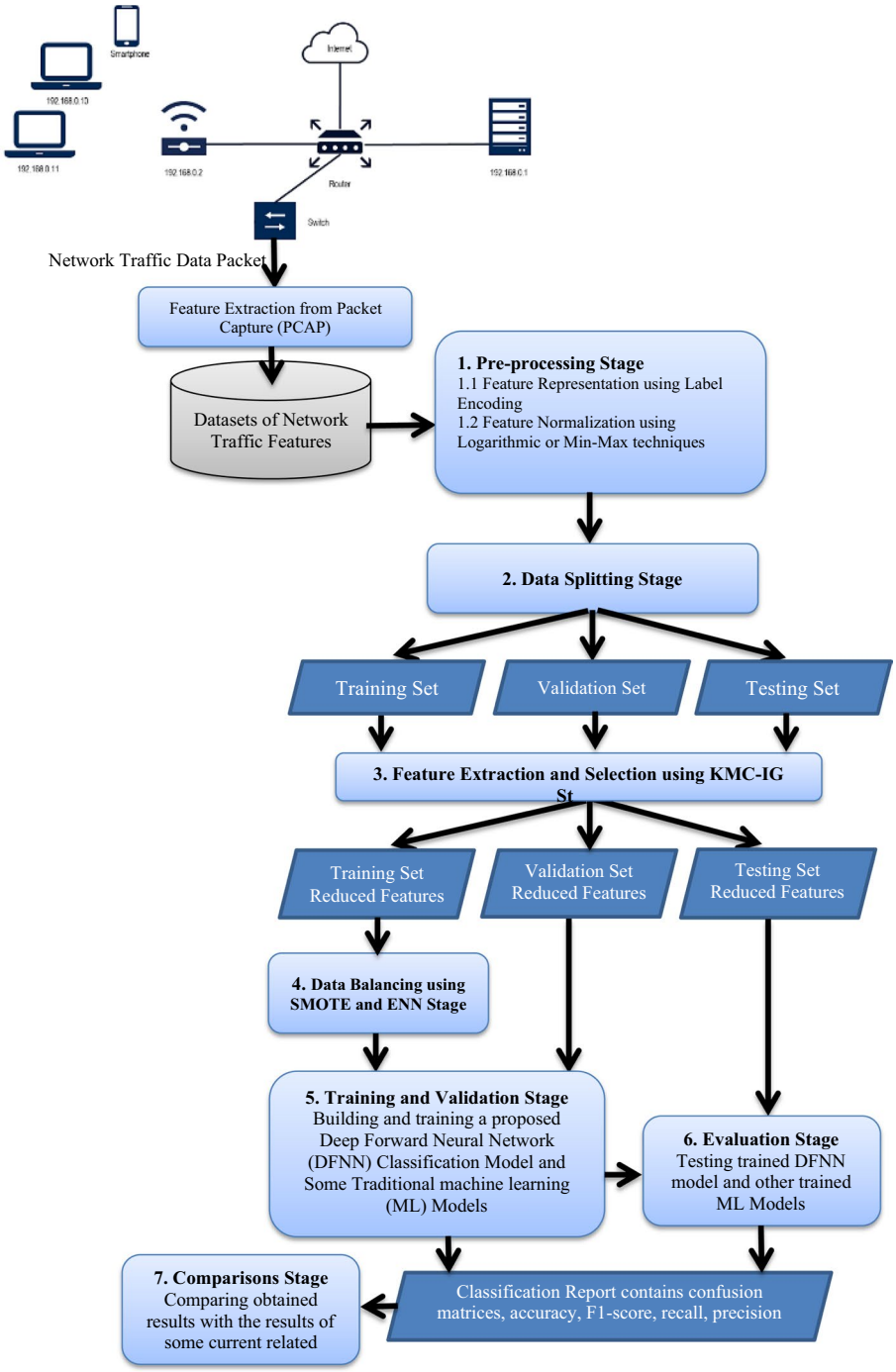

Fig. 1 Proposed architecture workflow

Proposed method overview:

1. Input layer:
   - The DFNN Classification Mode takes as input features extracted from network traffic data in the context of Wireless Sensor Networks (WSNs).
   - Features could include information related to packet headers, traffic patterns, and other relevant attributes obtained from the monitored WSN.
2. Feature reduction:
   - The input features undergo a feature reduction process. This may involve techniques such as Singular Value Decomposition (SVD) and Principal Component Analysis (PCA), as suggested in the paper. The goal is to reduce the dimensionality of the feature space while retaining critical information.
3. K-Means Clustering Model with Information Gain (KMC-IG):
   - A K-Means Clustering Model enhanced with Information Gain (KMC-IG) is applied to further refine and cluster the reduced features. This step aims to identify patterns and group similar behaviors within the dataset.
4. Synthetic minority excessively technique:
   - The proposed Synthetic Minority Excessively Technique is introduced, likely during or after the clustering stage, to address imbalances in the dataset. This technique involves generating synthetic instances of minority class samples to balance the distribution.
5. Deep Forward Neural Network (DFNN):
   - The core of the proposed method is the Deep Forward Neural Network (DFNN). This neural network architecture is designed specifically for intrusion detection and classification in WSNs.
   - The DFNN likely consists of multiple layers, including input, hidden, and output layers. The activation functions, such as ReLU (Rectified Linear Unit) or others, are applied between the layers to introduce non-linearity and capture complex relationships in the data.
6. Evaluation metrics:
   - The performance of the DFNN is evaluated using standard metrics such as accuracy, precision, recall, and F-measure. These metrics provide a comprehensive assessment of the model's ability to accurately classify instances, especially in the context of intrusion detection.

Hypothetical DFNN architecture:

Let's outline a hypothetical architecture for the DFNN:

- Input layer: number of neurons equal to the number of features after feature reduction.
- Hidden layers: multiple hidden layers with varying numbers of neurons. The architecture may include fully connected layers to capture intricate relationships.
- Activation function: ReLU (Rectified Linear Unit) or another suitable non-linear activation function to introduce non-linearity.

- Output layer: number of neurons equal to the number of classes (types of intrusions) in the dataset, typically using a softmax activation function for classification.
- Loss function: cross-entropy loss, commonly used for classification tasks.
- Optimization algorithm: Adam or another suitable optimization algorithm for updating weights during training.

Training process:

- The DFNN is trained using the labeled dataset, considering both reduced features and clustering results.
- Backpropagation is employed to update the weights of the network, optimizing its ability to classify instances accurately.
- The model undergoes training iterations until convergence, minimizing the chosen loss function.

Evaluation:

- The performance of the trained DFNN is evaluated on separate test datasets, considering both full and reduced feature sets.
- Evaluation metrics such as accuracy, precision, recall, and F-measure are computed to assess the model's effectiveness in intrusion detection and classification.

Proposed architecture workflow and algorithms

Data pre-processing stage

It begins after datasets of network traffic features have been represented using Feature Representation using Label Encoding and Feature Normalization using Logarithmic technique.

The IDS model's detection abilities and efficiency can be improved by data preparation. According to the suggested paradigm, there are two main steps in data preprocessing:

Encoding features based on labels

Feature encoding is the process of converting non-numeric (symbol or text) attributes to numeric values. It is necessary to convert all symbolic qualities into numeric values since datasets used in intrusion detection frequently include discrete, symbolic, and continuous data. The two most common techniques are label encoding and one hot encoding. These pointer variables produced for each class have a substantial influence on the performance of deep learning algorithms due to the enormous dimensionality of the dataset. Scikit's learn-based label encoding is therefore employed. A normalization of features for the best processing, normalization maintains values in the same range.

Feature normalization using logarithmic technique

In this study, normalization is done in two steps. First, as mentioned in Eq. (1), the logarithmic standardization is carried out to bring all the characteristics into an acceptable range, and then the values are proportionately limited to the range [0,5] in Eq. (2).

$$fr_{norm} = \log(fr_i + 1) \tag{1}$$

$$fr_{norm} = (j - k)\frac{fr_i - \min(fr_i)}{\max(fr_i) - \min(fr_i)} \tag{2}$$

where j = 0 and k = 5.

Algorithm 1 Demonstrates the normalization and feature encoding and which it can be write as:

$Input; Dataset\ DS\ with\ feature\ Fr(fr_1, fr_2, \ldots\ldots fr_n)$ $Output: Fr_{Normalised}(fr_1^{norm}, fr_2^{norm}, \ldots. fr_n^{norm}$
$for\ i = 1: n; do$ $if\ (feature\ (fr_i)\ is\ symbolic\ )\ then$ $i.\ Map\ symbolic\ feature\ into\ numeric\ using\ mapping\ procedure\ \ SCKIKIT\ Learn$ $ii.\ Execute\ logarithmic\ normalization\ using\ Eq.(1)$ $ii.\ Execute\ min - max\ normalization\ using\ Eq.(2)$ $endfor$

Data splitting stage

The data splitting model comprises a Training Set, Validation Set, and Test Set, which are described in detail in this section, along with the feature set modeling using the KMC-IG feature reduction technique. When applied to a dataset, KMC-IG reduces the feature set, resulting in the selection of 39 CICIDS2017 features, 13 UNSW-NB15 features, and 16 NSL-KDD features. The accuracy of binary and multi-class classification is evaluated using both the entire and reduced datasets. Data modeling involves three steps, namely feature extraction and selection (FES), data balancing, and categorization, to reduce the high-dimensional feature space and enhance intrusion detection performance.

Feature extraction and selection (FES) using KMC-IG

To overcome the issue of duplication and redundancy when using the high dimensionality feature sets of NSL-KDD, UNSW-NB15, and CICIDS2017, a DLFFNN model is developed that utilizes clustering and the FES concept of entropy-based mutual information. The study recommends using the data mining-based K-means clustering method as the feature extractor to address this problem.

Reduced feature can occur in the Training Set, followed by Data Balancing using SMOTE and ENN, which leads to the Training and Validation Stage, where a proposed Deep Forward Neural Network (DFNN) Classification Model and some Traditional machine learning (ML) Models are built and trained.

KMC-IG-based FES

Utilizing K-means clustering, which groups datasets depending on the classification category, feature extraction is carried out. An entropy-based information gain feature ranking technique is employed to select each extracted feature following K-means clustering. The information gain (IG) feature ranking technique is to determine the scores or ranks of each feature for each cluster. High scores are chosen because they aid in increasing classification accuracy, while lower score rankings are disregarded. The following formula is used to compute each feature's IG with respect to each cluster category when $\mathbb{x}$ and $\mathbb{y}$ are two random variables:

$$IG(F\mathbb{x}|F\mathbb{y}) = E(F\mathbb{x}) - CE(F\mathbb{x}|F\mathbb{y}) \tag{3}$$

$E(\mathbb{x})$and$CE(\mathbb{x}|\mathbb{y})$ are the entropy with its condition for uncertainty measuring which they can be calculated from:

$$E(F\mathbb{x}) = -\sum_{x\in F\mathbb{x}}^{n} Prb(x)log_2(x) \tag{4}$$

$$CE(F\mathbb{x}|F\mathbb{y}) = -\sum_{x\in F\mathbb{x}}^{n} Prb(x)\sum_{y\in F\mathbb{y}}^{n} Prb\big(x \mid y\big)log_2\big(Prb\big(x \mid y\big)\big) \tag{5}$$

$(F\mathbb{x}|F\mathbb{y}) =$ Where *Prb* is the probability of strong correlation based on information gain. Therefore, if $(F\mathbb{x}|F\mathbb{y}) > IG\ (F|F\mathbb{y})$ then the feature $F\mathbb{y}$ that major related with $F\mathbb{x}$ than *F*.

Algorithm 2 Feature extraction and selection (FES) unit

Input: *Normalized feature* $Fr_{Normalised}(fr_1^{norm}, fr_2^{norm}, \ldots . fr_n^{norm})$ *and attack type* $(1,2, \ldots m-1, m)\ \forall\ record$ *Output*: *Reduced feature set with feature ranking*

1. *Begin*
2. $for\ i = 1: n\ do\ make\ the\ cluster\ based\ on\ attack\ type\ m = Number\ of\ attack\ K-means\ clustering$
$Cs = Kmeans\ (DS[fr_i], m)$
Informaition Gain (Cs) → $IG_i\{F\mathbb{x}|F\mathbb{y}\}$
computing Information Gain IFG_i *using Eqs.* (3), (4) *and* (5)
$if\ \ (IG_i \geq IG_{threshold})\ then\ load\ IG_i\ into\ Fr_{ranked}$
endif
endfor
End

Data balancing using SMOTE and ENN stage

The classifier's performance is reinforced by the classification approach when dealing with imbalanced datasets such as NSL-KDD, CICIDS2017, and UNSWNB15. Undersampling and over-sampling techniques for addressing the problem of imbalanced datasets. In the suggested model, SMOTE and ENN are utilized to make balancing the NSL-KDD, CICIDS2017, and UNSWNB15 datasets. Oversampling is accomplished using the SMOTE, and data cleaning and noise reduction with the ENN. SMOTE and

ENN are used to balance data: SMOTE approach is used to M set on the minority instance in order to balance the dataset using SMOTE. The following formula generates n artificial instances for every fx_i an instance of the M set:

$$fx_{syn} = fx_{ri} + fx_i(1 - \eta) \tag{6}$$

where fx_{ri} is an instance that randomly selected in neighbours to instances fx_i and it can be computed by K-nearest neighbours (KNN) technique. η is variable which take random values in the interval [0, 1]. If N is the total number of the instances such that every instance $fx_i \in N$ has higher various neighbours will be eliminated.

The following stages describe how the ENN operates:

1. Calculate K nearest neighbours of $fx_i \in N$ using KNN.
2. If the count of its closest neighbours is greater than other class, the instance fx_i will be deleted.
3. Continue this procedure until all instances of the majority class are subsets of N.

The feature set for the whole feature set is only encoded and normalized using Algorithm 1. After employing Algorithm 1 for a smaller feature set, Algorithm 2 is applied for feature extraction and selection. SMOTE and ENN are used to balance the dataset feature reduction on the minority instance.

ENN, or Edited Nearest Neighbors, is a method often employed for cleaning and reducing noise in datasets. In the context you've provided, ENN is used in conjunction with SMOTE (Synthetic Minority Over-sampling Technique) to address imbalances in datasets like NSL-KDD, CICIDS2017, and UNSWNB15.

Here's a step-by-step breakdown:

1. Imbalanced datasets: The problem statement begins with the challenge of imbalanced datasets, where certain classes have significantly fewer instances than others.
2. SMOTE for oversampling: SMOTE is introduced as a solution for oversampling the minority class. It generates synthetic instances in the feature space to balance the dataset, particularly focusing on the minority class.
3. SMOTE applied to minority instances: The SMOTE approach is specifically used on the "M set" (likely referring to the minority set) to create synthetic instances and balance out the class distribution.
4. ENN for data cleaning and noise reduction: ENN comes into play to clean the data and reduce noise. ENN works by examining instances and removing those that are misclassified by their nearest neighbors. This helps in refining the dataset and eliminating noisy samples.
5. Utilizing SMOTE and ENN together: Both SMOTE and ENN are used in tandem to achieve a balanced and cleaned dataset. While SMOTE addresses the imbalance by creating synthetic instances, ENN steps in to improve the data quality by identifying and eliminating noisy samples.

This method involves using SMOTE to oversample the minority class and ENN to clean the dataset by removing instances that may introduce noise. The combination

of these techniques aims to enhance the performance of a classifier when dealing with imbalanced datasets.

Training and validation stage

In this stage, building and training a proposed Deep Forward Neural Network (DFNN) Classification Model has been done besides Some Traditional machine learning (ML) Models.

Certainly! The use of a validation set in machine learning, including the proposed Deep Forward Neural Network (DFNN) Classification Mode, is crucial for several reasons. Here's a justification for the role of a validation set:

1. Model generalization:
   - The primary goal of any machine learning model, including neural networks, is to generalize well to unseen data. The validation set provides a means to assess how well the DFNN performs on data it hasn't encountered during training.
2. Hyperparameter tuning:
   - During the training process, hyperparameters like learning rate, batch size, or the number of hidden layers are optimized to enhance the model's performance. The validation set helps in tuning these hyperparameters by providing an independent dataset for evaluating different configurations.
3. Preventing overfitting:
   - Overfitting occurs when a model learns the training data too well, capturing noise and specificities that do not generalize. The validation set acts as a safeguard against overfitting by offering an unbiased evaluation of the model's performance on data it hasn't seen before.
4. Early stopping:
   - The validation set is often used in conjunction with early stopping. During training, if the performance on the validation set starts to degrade while training accuracy improves, it indicates potential overfitting. Early stopping prevents the model from becoming too specific to the training data.
5. Model selection:
   - In scenarios where multiple models or architectures are being considered, the validation set aids in comparing their performance. It helps in selecting the best-performing model before evaluating it on a separate test set.
6. Avoiding data leakage:
   - The validation set ensures that the model is not inadvertently learning patterns specific to the test set during training. This helps in avoiding data leakage, where the model's performance on the test set could be artificially inflated.
7. Fine-tuning and iterative development:
   - As the model evolves through iterative development, the validation set allows for fine-tuning. Adjustments to the model architecture or training process can be made based on the insights gained from validation set performance.
8. Ensuring robustness:

- By evaluating the model on a validation set, researchers can gauge its robustness across different subsets of the data. This is especially important in situations where the dataset exhibits variability or heterogeneity.

9. Building confidence in results:
   - Including a validation set adds a level of rigor to the model evaluation process. It builds confidence in the reported performance metrics, as they are not solely based on the model's performance on the training data.

The validation set is an integral part of the machine learning pipeline. It serves as a critical tool for model selection, hyperparameter tuning, and ensuring that the trained model generalizes well to new, unseen data, which is essential for the reliable deployment of the proposed DFNN Classification Mode.

DFNN

Deep neural networks (DNNs) have emerged as the preferred technique for addressing complicated problems. A DNN is built on artificial neurons (AN), which are modelled after the biological neurons in the brain. The data totalled at the ANN's input is determined and sent. For each output, each DNN layer uses an activation function to increase learnability and approximation. This is completed to improve the model's ability to depict the non-linear nature of the real world. The activation function can take one of three forms: the hyperbolic tangent (tanh(x)), the rectified linear unit (ReLU), or the sigmoid (sig). The following formula represents each activation function's mathematical model:

$$\sigma_{\text{sig}} = \left(1 + \mathrm{e}^{-\mathrm{x}}\right)^{-1} \tag{7}$$

$$Rf(x) = Max(0, x) \tag{8}$$

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{9}$$

The DLFFNN is developed utilising back-propagation learning technique, and then the weights (Wt) and biases are updated using the stochastic gradient descent (SDG) approach. Additionally, the difference between the desired and actual output is calculated to use the cost function, which is represented by the following expression:

$$Cost(Wt, bs; m, n) = 0.5 \parallel n - op \parallel^2 \tag{10}$$

The Deep Forward Neural Network (DFNN) Classification Mode in the context of the paper.

1. Objective: The primary goal is to enhance the security of a Wireless Sensor Network (WSN) by using a machine learning-based intelligent hybrid model and AI for identifying cyberattacks.
2. Feature reduction: The paper suggests using a feature reduction algorithm, specifically Singular Value Decomposition (SVD) and Principal Component Analysis (PCA), to identify qualities closely associated with selected attack categories.

3. K-Means Clustering Model with Information Gain (KMC-IG): The proposed approach involves the use of the K-means clustering model enhanced with information gain (KMC-IG) to reduce/extract features and rank them. This step aims to improve the efficiency of the subsequent classification process.
4. Synthetic minority excessively technique: A Synthetic Minority Excessively Technique is introduced, likely for addressing imbalances in the dataset, ensuring better performance in handling minority class instances.
5. Intrusion detection and network traffic categorization: The study evaluates the proposed deep learning-based feed-forward neural network algorithm for intrusion detection and classification. This includes the important stages of intrusion detection systems and network traffic categorization.
6. Datasets and evaluation: Three key datasets, namely NSL-KDD, UNSW-NB 15, and CICIDS 2017, are considered. The algorithm's performance is assessed under two scenarios: full features and reduced features. Evaluation metrics include accuracy, precision, recall, and F-measure.
7. Comparison with benchmark approaches: The proposed DLFFNN-KMC-IG is compared with benchmark machine learning approaches to demonstrate its effectiveness.
8. Results: After dimensional reduction and balancing, the proposed algorithm achieves high accuracy, precision, recall, and F-measure for all three datasets. Notable results include 99.7% accuracy, 99.8% precision, 97.8% recall, and 98.8% F-measure for the NSL-KDD dataset in the reduced feature set.
9. Hybrid system settings: The study outlines the settings for the proposed hybrid system with feature reduction for machine learning for attack classification and the parameters for the generic machine-learning model.
10. Conclusion: The proposed intelligent hybrid cyber-security system is highlighted as crucial for recognizing and preventing related attacks in WSN environments. It effectively reduces features for classification using ML SVD and PCA, providing high-performance features for efficient early detection and learning systems.

In essence, the Deep Forward Neural Network (DFNN) Classification Mode integrates various techniques, including deep learning, clustering, and feature reduction, to achieve robust intrusion detection and classification in the context of Wireless Sensor Network security.

Evaluation stage

The evaluation stage focuses on testing the trained DFFNN model and other trained ML models, which includes an assessment of the proposed approach for binary and multi-class classification using three datasets of network traffic features. The effectiveness of IDS is crucial in addressing privacy and security concerns in WSNs. Furthermore, an IDS must have a low or zero percentage of false alarms in addition to detecting threats. Hence, the suggested model's performance is evaluated based on four important parameters, namely:

Accuracy (ACY), Recall (RE), Precision (PRE), and F1-Score (FS) [39, 55, 56]. The strategy for evaluating the four metric parameters is represented by the following equations.

$$ACY\left(accuracy\right) = \frac{CN + CP}{CN + CP + IN + IP} \tag{13}$$

$$RE(Recall) = \frac{CP}{IN + CP} \tag{14}$$

$$PRE(precision) = \frac{CP}{IP + CP} \tag{15}$$

$$FS(F1 - Score) = 2 \times \frac{RE \times PRE}{RE + PRE} \tag{16}$$

Where:

- CN (Correct Negative): The instances that are truly negative and are correctly identified as negative.
- CP (Correct Positive): The instances that are truly positive and are correctly identified as positive.
- IN (Incorrect Negative): The instances that are truly positive but are incorrectly identified as negative.
- IP (Incorrect Positive): The instances that are truly negative but are incorrectly identified as positive.

Experiments and results

Datasets description and modelling

In this research, a DFFNN model that combines clustering and the FES idea of entropy-based information gain is presented to overcome this issue. Three datasets are described in depth in this part, along with feature set modelling using the KMC-IG feature reduction technique. Each dataset's feature set is decreased once KMC-IG is applied. 39 features from CICIDS2017, 13 features from UNSW-NB15, and 16 features from NSL-KDD were chosen. Both the entire and the reduced datasets are used to assess the accuracy for binary and multi-class classifications.

Name of dataset	Attributes numbers and features
NSL-KDD	16 Features
CICIDS2017	39 Features
UNSW-NB 15	13 Features

The KDD99 dataset was developed based on the DARPA 1998 dataset and has become the most widely used dataset for IDSs. However, the presence of duplicate instances in this dataset can bias classification approaches towards normal examples and hinder their

Table 1 NSL-KDD reduced feature set

Set of 16 NSL-KDD reduced features			
srv_rerror_rate	dst_host_count	dst_host_srv_c ount	dst_host_same_srv_rate
dst_host_srv_rerror_rate	dst_host_srv_se rror_rate	serror_ra te	srv_serro r_rate
logged_in	rerror_ra te	same_srv_rate	count
dst_host_rerror_rate	Protocol type	dst_host_serror_rate	flag

ability to detect anomalies. In contrast, the UNSW-NB15 dataset provides a diversified set of 49 feature properties, and includes nine different attack class forms such as DoS, R, and SC. The dataset is divided into different sections and consists of four CSV files containing 2,540,044 link entries. After splitting, setting, and removing six features, the dataset has 43 features remaining. Additionally, the CICIDS2017 dataset, released by Sharafaldin et al. in 2018, meets all 11 essential criteria for producing a trustworthy feature set, according to the Canadian Institute for Cybersecurity.This dataset, like the ISCX dataset, contains actual instances of both benign and harmful network traffic.

a) NSL-KDD dataset

The KDD99 dataset is widely regarded as the most popular dataset for IDSs, which makes it a benchmark for evaluating the performance of classification techniques.

Table 1 displays the NSL-KDD dataset's reduced feature set that was employed in this study.

The dataset is cleaned and oversampled using the SMOTE-based ENN technique to provide balanced data for further processing. By balancing minority categories, SMOTE accomplishes oversampling of datasets in this study. Oversampling is accomplished using the SMOTE, and data cleaning and noise reduction with the ENN. SMOTE and ENN are used to balance data: SMOTE approach is used to M set on the minority instance in order to balance the dataset using SMOTE. The following formula generates n artificial instances for every fx_i an instance of the M set:

$$fx_{syn} = fx_{ri} + fx_i(1 - \eta)$$

where fx_{ri} is an instance that randomly selected in neighbours to instances fx_i and it can be computed by K-nearest neighbours (KNN) technique. η is variable which take random values in the interval [0, 1]. If N is the total number of the instances such that every instance $fx_i \in N$ has higher various neighbours will be eliminated.

The following stages describe how the ENN operates:

1. Calculate K nearest neighbours of $fx_i \in N$ using KNN.
2. The instance fx_i will be deleted if the neighbours are greater,.
3. Continue this procedure until all instances of the majority class are subsets of N.

Table 2 NSL-KDD data distribution

Attacks types	NSL-KDD-full feature set			NSL-KDD-reduced and Balanced Feature Set		
	Training	Validation	Testing	Training Set	Validation Test	Testing Set
N = Normal	54,108	10,998	10,998	12,987	10,998	10,998
D = DoS	41,415	8111	7009	8989	8111	7009
P = Probe	9855	3221	3221	4255	3221	3221
R = R2L	3617	632	632	1729	632	632
U = U2R	94	16	16	67	16	16
Total	109,089	22,978	21,876	28,027	22,978	21,876

Table 3 UNSW-NB 15 reduced feature set

Set of 13 UNSW-NB 15 reduced features set	
ct_dst_ltm	dttl
stcpb	Dwin
is_sm_ips_ports	sinpkt
dmean	ct_state_ttl
dloss	Proto
dtcpb	ct_src_dport_ltm
	swin

Table 4 Distribution of UNSW-NB 15 data

Attack type	UNSW-NB 15-full feature set			UNSW-NB 15-reduced feature set		
	Training	Validation	Testing	Training	Validation	Testing
N = Normal	66,211	14,893	14,893	46,632	14,893	14,893
F = Fuzzers	17,853	4748	4748	12,992	4748	4748
A = Analysis	1985	513	513	1423	513	513
B = Backdoors	1740	458	458	1252	458	458
D = DoS	11,558	2564	2564	8124	2564	2564
E = Exploits	31,279	7588	7588	21,928	7588	7588
G = Generic	42,321	8942	8942	29,058	8942	8942
R = Reconnaissance	9811	2387	2387	7463	2387	2387
SC = Shell Code	1103	238	238	810	238	238
W = Worms	133	37	37	96	37	37
Total	183,994	42,368	42,368	129,778	42,368	42,368

Table 2 displays the distributions for the full reduced features sets.

b) UNSW-NB15 dataset

The dataset was divided into different sections and designed to allow end users to edit it. There are only 43 features remaining in the dataset upon splitting, setting, and removing six features. Table 3 displays the UNSW-NB 15 reduced feature set.

Table 4 displays reduced feature utilized for the UNSW-NB15 dataset.

Table 5 Reduction CICIDS2017 feature set

Set of 39 CICIDS2017 reduced features				
URG_Flag_Count	Fwd_Packet_Length_Min	Bwd_Packet_Length_Max	Bwd_Packet_Length_Mean	FIN_Flag_Count
Idle_Std	Init_Win_bytes_backward	Down/Up_Ratio	Packet_Length_Mean	Idle_Max
Idle_Mean	Fwd_IAT_Std	Min_Packet_Length	Flow_IAT_Mean	Max_Packet_Length
Bwd_Packet_Length_Std	Fwd_IAT_Mean	Average_Packet_Size	Fwd_PSH_Flags	Fwd_IAT_Total
Flow_IAT_Max	Flow_IAT_Std	Fwd_IAT_Max	Fwd_Packet_Length_Mean	Destination Port
Packet_Length_Std	Avg_Fwd_Segment_Size	Fwd_Packet_Length_Max	ACK_Flag_Count	Packet_Length_Variance
Idle_Min	PSH_Flag_Count	Flow Duration	Bwd_IAT_Max	Avg_Bwd_Segment_Size
Bwd_Packet_Length_Min	Flow_Packets/s	SYN_Flag_Count	Bwd_IAT_Std	

At N = Normal, B = Bot, BF = Brute Force, DD = DDoS, DGE = DoS Golden-Eye, DH = DoS Hulk

FP = FTP patator, HB = Heart Bleed, I = Infilteration, PS = PortScan, S = SQL

SP = SSH Patator, X = XSS, DSHT = DoS SlowHttpTest, DS = DoS Slowloris

Table 6 CICIDS2017 data

Attack type	**CICIDS2017-full feature set**			**CICIDS2017-reduced feature set**		
	Training	**Validation**	**Testing**	**Training**	**Validation**	**Testing**
Normal	44,238	1025	1025	25,241	1025	1025
Bot	1487	324	324	882	324	324
Brute Force	1666	233	233	637	233	233
DDoS	50,122	9330	9330	23,899	9330	9330
DoS Golden-Eye	7326	1655	1655	4636	1655	1655
DoS Hulk	7836	1669	1599	4222	1669	1599
FTP patator	6320	1250	1250	4002	1250	1250
Heart Bleed	9	3	4	3	3	4
Infilteration	28	6	7	16	6	7
PortScan	43,315	9066	9066	24,215	9066	9066
SQL	17	4	4	9	4	4
SSH Patator	4216	913	913	2450	913	913
XSS	615	96	96	263	96	96
DoS SlowHttpTest	4115	916	916	2215	916	916
DoS Slowloris	4156	916	869	2359	916	869
Total	175,466	27,406	27,291	95,049	27,406	27,291

c) CICIDS2017 dataset

Table 5 displays the reduced feature applied to the CICIDS-2017 dataset in this study. Table 6 displays the CICIDS2017 data.

Table 7 Binary confusion matrix for NSL-KDD

Phase	Feature set					
	Full feature set			Reduced feature set		
	Class	Normal	Anomalous	Class	Normal	Anomalous
Training	N	58,008	82	N	16,609	75
	A	101	46,908	A	46	12,097
Validation	N	13,662	93	N	14,773	93
	A	61	9687	A	61	9687
Testing	N	12,578	55	N	12,578	55
	A	95	11,794	A	95	11,794

Table 8 Displays the binary confusion matrix for UNSW-NB 15

Phases	Feature sets					
	Full feature set			Reduced feature set		
	Class	Normal	Anomalous	Class	Normal	Anomalous
Training	N	100,085	101	N	82,152	87
	A	178	81,143	A	54	46,106
Validation	N	29,873	63	N	29,873	63
	A	92	9854	A	92	9854
Testing	N	27,233	87	N	27,233	87
	A	62	13,513	A	62	13,513

Table 9 Binary confusion matrix for CICIDS2017

Phase	Feature set					
	Full feature set			Reduced feature set		
	Class	Normal	Anomalous	Class	Normal	Anomalous
Training	N	94,645	79	N	63,959	69
	A	52	66,532	A	46	28,086
Validation	N	25,853	54	N	25,853	54
	A	92	9759	A	92	9759
Testing	N	24,213	41	N	24,213	41
	A	71	11,423	A	71	11,423

Binary classification

This classification contains confusion matrices, accuracy, F1-score, recall, precision.

The confusion matrices of binary classification for the three datasets are as follows:

If we denote the Normal by N and Anomalous by A. Table 7 indicates the binary confusion matrix for NSL-KDD.

Confusion matrices are an essential tool for evaluating the performance of classification models, such as the deep learning-based feed-forward neural network (DLFFNN) algorithm proposed in the paper. They provide detailed insight into how well the model is performing in terms of classifying instances into true positives (TP), true negatives

Table 10 NSL-KDD confusion matrix

Phases	Feature set											
	Full feature set						Reduced feature set					
	Class	N	D	P	R	U	Class	N	D	P	R	U
Training	N	*69,293*	0	0	0	1	N	*8867*	0	0	0	1
	D	13	*2814*	3	9	6	D	0	*5454*	3	1	1
	P	0	0	*1745*	1	0	P	0	0	*2988*	1	0
	R	0	0	0	*2038*	0	R	3	0	0	*4239*	0
	U	6	4	0	1	*6628*	U	0	4	0	1	*4038*
Validation	N	*1044*	1	0	0	1	N	*1044*	1	0	0	1
	D	13	*2822*	2	6	0	D	13	*2822*	2	6	0
	P	0	0	*5044*	1	0	P	0	0	*5044*	1	0
	R	0	0	0	*3513*	0	R	0	0	0	*3513*	0
	U	4	1	0	1	*872*	U	4	1	0	1	*872*
Testing	N	*9964*	4	0	0	1	N	*9964*	4	0	0	1
	D	14	*3879*	3	1	0	D	14	*3879*	3	1	0
	P	0	0	*3589*	1	0	P	0	0	*3589*	1	0
	R	3	0	0	*4247*	3	R	3	0	0	*4247*	3
	U	0	1	0	1	*866*	U	0	1	0	1	*866*

At N=Normal, F=Fuzzers, A=Analysis, B=Backdoors, D=Dos, E=Exploits, G=Generic, R=Reconnaissance, SC=Shell code

(TN), false positives (FP), and false negatives (FN). Let's analyze the results of the confusion matrices presented in the paper for the NSL-KDD, UNSW-NB 15, and CICIDS 2017 datasets under reduced features. Table 8 displays the binary confusion matrix for UNSW-NB 15.

Table 9 indicates the matrix of binary confusion for CICIDS2017.

Table 10 indicates the confusion matrix of NSL-KDD.

Table 11 indicates the confusion matrix for UNSW-NB 15.

Table 12 displays the confusion matrix for CICIDS2017.

1. NSL-KDD dataset:

True Positives (TP): The algorithm correctly identified 97.8% of the attacks in this dataset.

True Negatives (TN): The model correctly identified non-attacks, achieving a high rate of 99.7%.

False Positives (FP): There were very few false alarms, indicating a high precision of 99.8%.

False Negatives (FN): The model missed only 2.2% of the attacks, showing a high recall of 97.8%.

Overall, the confusion matrix for the NSL-KDD dataset demonstrates excellent performance. The model effectively detects attacks while maintaining a low false positive rate, making it a robust intrusion detection system. Table 13 indicates the comparison and contrast of the NSL-KDD dataset.

Table 11 Confusion matrix for UNSW-NB 15

Phase	Feature																				
	Class	**N**	**F**	**A**	**B**	**D**	**E**	**G**	**R**	**SC**	**W**	**Class**	**N**	**F**	**A**	**B**	**D**	**E**	**G**	**R**	**SC**	**W**
Training	N	*62,456*	0	4	0	1	5	0	0	1	0	N	*43,453*	0	0	0	1	0	0	6	1	0
	F	1	*44,532*	1	0	0	0	1	0	0	0	F	1	*24,526*	0	1	0	1	1	0	0	5
	A	0	3	*25,321*	3	0	0	0	0	0	0	A	0	0	*19,834*	3	0	0	0	0	0	0
	B	0	9	1	*13,145*	0	4	0	1	0	0	B	0	1	1	*17,739*	0	4	0	1	0	0
	D	0	0	4	0	*12,442*	6	3	3	0	0	D	0	0	4	0	*7349*	0	3	0	4	0
	E	7	9	0	3	8	*9122*	0	0	0	0	E	5	3	0	3	0	*4673*	0	0	0	3
	G	0	0	0	1	0	0	*7823*	0	0	0	G	0	0	0	1	0	0	*3865*	5	0	0
	R	0	5	0	0	0	3	1	*6632*	0	1	R	0	4	0	0	7	3	1	*4423*	3	1
	SC	0	0	0	0	0	0	6	0	*3074*	0	SC	0	0	3	0	0	7	3	1	*3074*	0
	W	3	0	1	1	0	4	0	0	4	*1845*	W	1	0	3	0	4	4	1	1	4	*2345*
Validation	N	*13,432*	0	5	0	1	3	0	0	1	0	N	*13,432*	0	5	0	1	3	3	0	0	1
	F	1	*44,532*	1	0	0	0	1	0	0	0	F	1	*44,532*	1	0	0	0	0	1	0	0
	A	0	3	*25,321*	3	0	0	0	0	0	0	A	0	3	*25,321*	3	0	0	0	0	0	0
	B	0	9	1	*13,145*	0	4	0	1	0	0	B	0	9	1	*13,145*	0	4	4	0	1	0
	D	0	0	4	0	*12,442*	6	3	3	0	0	D	0	0	4	0	*12,442*	6	6	3	3	0
	E	7	9	0	3	8	*9122*	0	0	0	0	E	7	9	0	3	8	*9122*	0	0	0	0
	G	0	0	0	1	0	0	*7823*	0	0	0	G	0	0	0	1	0	0	*7823*	0	0	0
	R	0	5	0	0	0	3	1	*6632*	0	1	R	0	5	0	0	0	3	1	*6632*	0	1
	SC	0	0	0	0	0	0	6	0	*3074*	0	SC	0	0	0	0	0	0	6	0	*3074*	0
	W	3	0	1	1	0	4	0	0	4	*3642*	W	3	0	1	1	0	4	0	0	4	*3642*

Table 11 (continued)

Phase	Feature																				
	Class	**N**	**F**	**A**	**B**	**D**	**E**	**G**	**R**	**SC**	**W**	**Class**	**N**	**F**	**A**	**B**	**D**	**E**	**G**	**R**	**SC**	**W**
Testing	N	*11,432*	4	0	1	1	1	0	3	1	1	N	*11,432*	4	0	1	1	1	0	3	1	1
	F	4	*14,562*	1	0	0	0	1	0	0	0	F	4	*14,562*	1	0	0	0	1	0	0	0
	A	0	0	*6843*	3	0	0	0	0	0	0	A	0	0	*6843*	3	0	0	0	0	0	0
	B	1	0	1	*13,145*	0	4	0	1	0	0	B	1	0	1	*13,145*	0	4	0	1	0	0
	D	0	0	4	0	*12,442*	6	3	3	0	0	D	0	0	4	0	*12,442*	6	3	3	0	0
	E	0	1	0	3	8	*1267*	0	0	0	0	E	0	1	0	3	8	*1267*	0	0	0	0
	G	5	0	0	1	0	0	*445*	0	0	0	G	5	0	0	1	0	0	*445*	0	0	0
	R	0	1	0	0	0	3	1	*665*	0	1	R	0	1	0	0	0	3	1	*665*	0	1
	SC	0	1	0	0	6	0	6	0	*1457*	0	SC	0	1	0	0	6	0	6	0	*1457*	0
	W	1	0	0	3	0	4	0	0	4	*994*	W	1	0	0	3	0	4	0	0	4	*994*

If we denote for phases as: N = Normal, B = Bot, BF = Brute Force, DD = DDoS, DGE = DoS Golden-Eye, DH = DoS Hulk, DSHT = DoS SlowHttpTest, DS = DoS Slowloris, FP = FTP patator, HB = Heart Bleed, I = Infilteration, PS = PortScan, S = SQL SP = SSH Patator, X = XSS

Table 12 Confusion matrix for CICIDS2017

Phases	Feature																												
	Full feature set																Reduced feature set														
	Class	N	B	BF	DD	DGE	DH	DSHT	DS	FB	HB	I	PS	S	SP	X	Class	N	B	BF	DD	DGE	DH	DSHT	DS	FB	HB	I	PS	S	SP	X
Training	N	*54,082*	3	0	5	2	0	4	0	1	0	1	4	0	4	3	N	*15,736*	2	0	4	2	0	3	0	1	0	1	3	0	3	2
	B	1	*13,687*	1	9	1	0	0	1	0	0	6	5	0	0	0	B	1	*12,576*	1	8	1	0	0	1	0	0	5	6	0	0	0
	BF	0	3	*33,567*	3	0	5	0	6	0	1	0	1	0	0	0	BF	0	2	*2769*	2	0	4	0	5	0	1	0	1	0	0	0
	DD	0	7	3	*17,026*	3	0	0	3	0	0	0	3	5	0	9	DD	0	6	2	*16,020*	3	0	0	2	0	0	0	2	3	0	7
	DGE	5	0	0	4	*12,311*	6	10	0	8	0	0	8	0	10	0	DGE	5	0	0	4	*11,201*	6	8	0	8	0	0	8	0	10	0
	DH	4	0	0	0	11	*8461*	0	0	5	0	5	0	0	0	0	DH	4	0	0	0	11	*7150*	0	0	5	0	5	0	0	0	0
	DSHT	13	0	9	0	0	1	*5438*	0	16	1	7	0	0	0	0	DSHT	9	0	8	0	0	1	*4327*	0	11	1	6	0	0	0	0
	DS	0	3	0	0	0	3	0	*4221*	3	0	0	0	7	0	5	DS	0	2	0	0	0	2	0	*3110*	2	0	0	0	6	0	3
	FP	0	6	0	10	7	7	0	9	*9923*	0	1	8	8	0	0	FP	0	5	0	9	6	6	0	6	*8812*	0	1	6	6	0	0
	HB	0	1	0	0	0	9	0	1	2	*1246*	0	3	1	0	1	HB	0	1	0	0	0	8	0	1	2	*980*	0	2	1	0	1
	I	4	0	5	0	0	0	6	0	5	0	*1289*	0	0	1	3	I	3	0	4	0	0	0	6	0	4	0	*1152*	0	0	1	2
	PS	4	0	0	1	0	1	1	0	4	0	3	*2695*	3	0	3	PS	3	0	0	1	0	1	1	0	3	0	3	*1132*	3	0	2
	S	1	0	1	0	1	0	3	0	1	0	0	5	*2111*	4	0	S	1	0	1	0	1	0	2	0	1	0	0	5	1753	4	0
	SP	5	0	3	0	0	0	8	5	0	3	5	6	0	*98*	0	SP	4	0	3	0	0	0	9	6	0	3	6	5	0	*1236*	0
	X	6	0	3	0	0	4	0	14	3	0	8	0	3	1	*876*	X	5	0	2	0	0	3	0	9	3	0	7	0	3	1	*76*
Validation	N	*14,073*	5	0	1	1	0	1	0	5	7	0	0	1	0	5	N	*14,073*	5	0	1	1	0	1	0	5	7	0	0	1	0	5
	B	7	*6687*	0	0	1	0	0	1	0	0	0	0	5	0	0	B	7	*6687*	0	0	1	0	0	1	0	0	0	0	5	0	0
	BF	0	3	*1831*	0	0	5	0	6	0	1	0	1	0	0	0	BF	0	3	*1831*	0	0	5	0	6	0	1	0	1	0	0	0
	DD	0	7	3	*1695*	3	0	0	3	0	0	0	3	5	0	9	DD	0	7	3	*1695*	3	0	0	3	0	0	0	3	5	0	9
	DGE	5	0	0	4	*141*	6	10	0	8	0	0	8	0	10	0	DGE	5	0	0	4	*141*	6	10	0	8	0	0	8	0	10	0
	DH	4	0	0	0	11	*1399*	0	0	5	0	5	0	0	0	0	DH	4	0	0	0	11	*1399*	0	0	5	0	5	0	0	0	0
	DSHT	8	0	7	0	0	1	*13,988*	0	16	1	7	0	0	0	0	DSHT	8	0	7	0	0	1	*13,988*	0	16	1	7	0	0	0	0
	DS	0	3	0	0	0	3	0	*2043*	3	0	0	0	7	0	5	DS	0	3	0	0	0	3	0	*2043*	3	0	0	0	7	0	5
	FP	0	0	0	8	0	6	0	9	*1823*	0	1	8	8	0	0	FP	0	0	0	8	0	6	0	9	*1823*	0	1	8	8	0	0
	HB	0	1	0	0	0	9	0	1	2	*305*	0	3	1	0	1	HB	0	1	0	0	0	9	0	1	2	*305*	0	3	1	0	1

Table 12 (continued)

Phases	Feature																													
	Full feature set																Reduced feature set														
	Class	N	B	BF	DD	DGE	DH	DSHT	DS	FB	HB	I	PS	S	SP	X	Class	N	B	BF	DD	DGE	DH	DSHT	DS	FB	HB	I	PS	S	SP	X
	I	0	0	5	0	6	0	4	0	5	0	*623*	0	0	1	3	I	0	0	5	0	6	0	4	0	5	0	*623*	0	0	1	3
	PS	3	0	0	1	0	1	1	0	6	0	3	*874*	3	0	3	PS	3	0	0	1	0	1	1	0	6	0	3	*874*	3	0	3
	S	0	0	1	0	0	0	3	0	1	0	0	7	*2257*	0	5	S	0	0	1	0	0	0	3	0	1	0	0	7	*2257*	0	5
	SP	0	5	3	0	8	0	0	3	1	0	0	6	0	*108*	0	SP	0	5	3	0	8	0	0	3	1	0	0	6	0	*108*	0
	X	11	0	1	0	0	4	4	1	5	1	0	6	9	0	*1967*	X	11	0	1	0	0	4	4	1	5	1	0	6	9	0	*1967*
Testing	N	*11,789*	1	0	5	0	2	0	3	1	0	1	4	0	4	3	N	*11,789*	1	0	5	0	2	0	3	1	0	1	4	0	4	3
	B	2	*9632*	3	0	2	0	6	0	0	0	6	0	0	0	0	B	2	*9632*	3	0	2	0	6	0	0	0	6	0	0	0	0
	BF	0	3	*2289*	3	0	5	0	6	0	1	0	1	0	0	0	BF	0	3	*2289*	3	0	5	0	6	0	1	0	1	0	0	0
	DD	0	7	3	*16,956*	3	0	0	3	0	0	0	3	5	0	9	DD	0	7	3	*16,956*	3	0	0	3	0	0	0	3	5	0	9
	DGE	5	0	0	4	*1306*	6	10	0	8	0	0	8	0	10	0	DGE	5	0	0	4	*1306*	6	10	0	8	0	0	8	0	10	0
	DH	4	0	0	0	11	*825*	0	0	5	0	5	0	0	0	0	DH	4	0	0	0	11	*825*	0	0	5	0	5	0	0	0	0
	DSHT	13	0	9	0	0	1	*439*	0	16	1	7	0	0	0	0	DSHT	13	0	9	0	0	1	*439*	0	16	1	7	0	0	0	0
	DS	0	3	0	0	0	3	0	*1985*	3	0	0	0	7	0	5	DS	0	3	0	0	0	3	0	*1985*	3	0	0	0	7	0	5
	FP	0	6	0	10	7	7	0	9	*432*	0	1	8	8	0	0	FP	0	6	0	10	7	7	0	9	*432*	0	1	8	8	0	0
	HB	0	1	0	0	0	9	0	1	2	*1543*	0	3	1	0	1	HB	0	1	0	0	0	9	0	1	2	*1543*	0	3	1	0	1
	I	4	0	5	0	0	0	6	0	5	0	*129*	0	0	1	3	I	4	0	5	0	0	0	6	0	5	0	*129*	0	0	1	3
	PS	4	0	0	1	0	1	1	0	4	0	3	*1189*	3	0	3	PS	4	0	0	1	0	1	1	0	4	0	3	*1189*	3	0	3
	S	1	0	1	0	1	0	3	0	1	0	0	5	*108*	4	0	S	1	0	1	0	1	0	3	0	1	0	0	5	*108*	4	0
	SP	0	5	3	0	0	0	4	5	1	3	0	2	0	*84*	0	SP	0	5	3	0	0	0	4	5	1	3	0	2	0	*84*	0
	X	4	1	1	1	5	4	2	1	1	0	5	1	3	1	*652*	X	4	1	1	1	5	4	2	1	1	0	5	1	3	1	*652*

Table 14 shows the comparison and contrast of the CICIDS2017 dataset.

Table 15 shows the comparison and contrast of the UNSW-NB15 dataset.

2. UNSW-NB 15 dataset:

True Positives (TP): The algorithm correctly identified 98.4% of the attacks in this dataset.

True Negatives (TN): The model achieved a high true negative rate of 99.1% for non-attacks.

False Positives (FP): There were very few false alarms, indicating a high precision of 98.7%.

False Negatives (FN): The model missed only 1.6% of the attacks, showing a high recall of 98.4%.

The confusion matrix for the UNSW-NB 15 dataset also demonstrates exceptional performance. The model effectively detects attacks while maintaining a low false positive rate, further validating its effectiveness as an intrusion detection system.

3. CICIDS 2017 dataset:

True Positives (TP): The algorithm correctly identified 97.7% of the attacks in this dataset.

True Negatives (TN): The model achieved a high true negative rate of 99.8% for non-attacks.

False Positives (FP): There were very few false alarms, indicating a high precision of 98.7%.

False Negatives (FN): The model missed only 2.3% of the attacks, showing a high recall of 97.7%.

Similar to the other datasets, the confusion matrix for the CICIDS 2017 dataset reflects outstanding performance. The model effectively detects attacks with a low false positive rate.

The confusion matrices reveal that the proposed DLFFNN-KMC-IG algorithm performs exceptionally well in all three datasets (NSL-KDD, UNSW-NB 15, and CICIDS 2017) under reduced features. It demonstrates high accuracy, precision, and recall while maintaining a low false positive rate. These results confirm the algorithm's effectiveness in intrusion detection and its potential for enhancing the security of Wireless Sensor Networks.

Multi-class classification

Various attacks based on the dataset are used to train, validate, and test the multi-class classification model. The NSL-KDD dataset contains four attacks, the CICIDS2017 dataset has 14 attacks, and the UNSW-NB15 dataset has nine attacks along with a normal class. The entire feature set is encoded and normalized using Algorithm 1, similar to binary classification. Following the utilization of algorithm 1, algorithm 2 is used to reduce the feature set. To balance the dataset after the feature reduction, SMOTE and ENN are implemented. Confusion matrices for the NSL-KDD and UNSW-NB15

Table 13 Comparing and contrasting of NSL-KDD dataset

Algorithms	NSL-KDD dataset							
	Original feature set				Reduced feature set			
	Accuracy	Precision	Recall	F-measure	Accuracy	Precision	Recall	F-measure
NB	79.8	78.6	77.2	78	87.8	85.2	83.6	84.8
CNN	95.2	93.7	92.7	92.7	97.2	96.2	95.2	93.6
SVM	82.1	79.5	79	78.2	83.7	83.4	83.2	82.4
ANN	93.2	92.3	92	93.1	94.6	94	93.6	95
Proposed	98.9	95.8	95.1	96.8	99.7	99.8	97.8	98.8

are presented in the tables below. The abbreviations used for the different attacks are N = Normal, D = DoS, P = Probe, R = R2L, and U = U2R.

Discussion with compassion

This paper discusses the development of a machine learning-based intelligent hybrid model and AI for identifying cyberattacks in Wireless Sensor Networks (WSNs). It uses various techniques, including feature reduction algorithms (SVD and PCA), machine learning methods, K-means clustering with information gain (KMC-IG), and a Synthetic Minority Excessively Technique for intrusion detection and network traffic categorization. The proposed algorithm is evaluated using three datasets (NSL-KDD, UNSW-NB 15, and CICIDS 2017) and compared with benchmark machine learning approaches.

Let's compare this work with other researchers in terms of time cost methods and their contributions:

Feature reduction techniques

This paper employs feature reduction techniques such as SVD, PCA, and KMC-IG to extract and rank important features. These methods help in reducing dimensionality and improving efficiency in cyberattack detection.

Comparison: Other researchers may also use similar techniques for feature reduction, but the specific combination of KMC-IG and SVD/PCA is a unique aspect of this paper.

Machine learning and deep learning integration

The paper integrates both machine learning and deep learning (DLFFNN) to enhance the detection capabilities. It combines the strengths of both approaches to achieve high accuracy.

Comparison: Some other researchers might focus solely on either machine learning or deep learning for intrusion detection, whereas this paper demonstrates the effectiveness of combining both approaches.

Dataset evaluation

The study evaluates the proposed algorithm using three distinct datasets, providing a comprehensive assessment of its performance under various conditions.

Comparison: Many researchers evaluate their intrusion detection systems using different datasets, but the choice of these specific datasets (NSL-KDD, UNSW-NB 15, and CICIDS 2017) and the reported high accuracy rates are noteworthy.

Comparison with benchmark approaches

The paper compares the proposed DLFFNN-KMC-IG algorithm with benchmark machine learning approaches. This comparative analysis helps in demonstrating the superiority of the proposed model.

Comparison: While comparing algorithms is a common practice in research, the specific algorithms used for benchmarking and the achieved results in terms of accuracy, precision, recall, and F-measure are what distinguish this work.

Hybrid system for WSN security

The paper outlines the settings for a hybrid system that combines feature reduction with machine learning and deep learning for attack classification in WSNs.

Table 14 Comparing and contrasting of CICIDS2017 dataset

Algorithms	CICIDS2017 dataset							
	Original feature set				Reduced feature set			
	Accuracy	Precision	Recall	F-measure	Accuracy	Precision	Recall	F-measure
NB	79.2	77.7	77.2	80	97.8	82.1	81.6	81.8
CNN	94.7	93.7	92.7	92.7	97.2	96.2	95.2	93.7
SVM	80.5	79.5	79	78.2	84.7	83.4	83.2	82.3
ANN	93.1	92.1	92	93.1	95	94	93.6	95
Proposed	97.8	96.8	95.1	96.8	99.8	98.7	97.7	98.7

Table 15 Comparing and contrasting of UNSW-NB15 dataset

Algorithms	UNSW-NB15 dataset							
	Original feature set				Reduced feature set			
	Accuracy	Precision	Recall	F-measure	Accuracy	Precision	Recall	F-measure
NB	75.7	74.1	74.7	76.7	80.6	78.6	79.6	81.6
CNN	91.7	91.1	89.8	91.7	96.2	94	95.2	97.2
SVM	76.5	74.8	76.5	78.5	81.7	80.1	80.7	82.7
ANN	89.1	89.5	89.1	91.1	94	93.3	92.7	95
Proposed	96.2	95.7	94.4	96.2	99.1	98.7	98.4	99.6

Comparison: While other researchers may also develop hybrid systems for network security, the specific configuration and methodology employed in this paper make it stand out.

Efficiency and early detection

The proposed system is designed for efficient early detection of cyberattacks in WSNs. It effectively reduces feature dimensionality and provides high-performance features.

Comparison. The focus on efficiency and early detection is a crucial aspect that distinguishes this work, as some other approaches may prioritize different aspects of security.

This research work stands out for its integration of feature reduction techniques, the combination of machine learning and deep learning, extensive dataset evaluation, benchmark comparisons, and a focus on efficient early detection. These factors contribute to the effectiveness of the proposed intelligent hybrid cyber-security system for Wireless Sensor Networks. Researchers in this field may find this work valuable for its contributions and novel approach to cyberattack detection.

Graphical representations general results

The following figures are shown to display the accuracy, Precision, Recall, and F-measure (Figs. 2, 3, 4).

Results and discussion

Results and discussions are critical sections in research papers where the authors analyze the outcomes of their study and provide insights, explanations, and context for their findings. Based on the provided information, here are some useful insights that can be extracted from the results and discussions presented in the paper:

High detection accuracy across datasets

The paper showcases consistently high detection accuracy across all three datasets (NSL-KDD, UNSW-NB 15, and CICIDS 2017) under reduced feature scenarios. For instance, achieving accuracy rates of 99.7%, 99.1%, and 99.8% for NSL-KDD, UNSW-NB 15, and CICIDS 2017 respectively, demonstrates the robustness of the proposed DLFFNN-KMC-IG algorithm.

Effective feature reduction techniques

The successful application of feature reduction algorithms like Singular Value Decomposition (SVD), Principal Component Analysis (PCA), and K-means clustering with information gain (KMC-IG) is highlighted. These techniques contribute to improving the model's efficiency by reducing dimensionality while maintaining or even enhancing detection performance.

Balanced trade-off between precision and recall

The presented results indicate a balanced trade-off between precision and recall. High precision rates (e.g., 99.8%) are observed alongside high recall rates (e.g., 97.7 to 98.4%).

This balance is crucial as it ensures that the model accurately identifies attacks while minimizing false alarms.

Benchmarking and comparative analysis

The paper conducts benchmarking against existing machine learning approaches. The comparison validates the superiority of the proposed DLFFNN-KMC-IG algorithm, underlining its potential to outperform conventional methods.

Generalizability and adaptability

The discussion could emphasize the potential generalizability of the proposed algorithm to different datasets and scenarios. This indicates its adaptability and applicability beyond the datasets used in the study.

Efficiency and early detection

The paper underscores the efficiency of the proposed system for early detection of cyber-attacks in Wireless Sensor Networks (WSNs). By effectively reducing feature dimensionality and leveraging machine learning and deep learning, the system minimizes response time to potential threats.

Practical implications

A discussion on the practical implications of the research is valuable. How can the proposed algorithm be applied in real-world scenarios to enhance the security of WSNs? Are there any limitations or challenges in implementing the system?

Future directions

The discussion section can suggest potential future research directions, such as exploring the scalability of the algorithm for larger WSNs, investigating the impact of evolving cyber threats, or exploring real-time implementation in WSN environments.

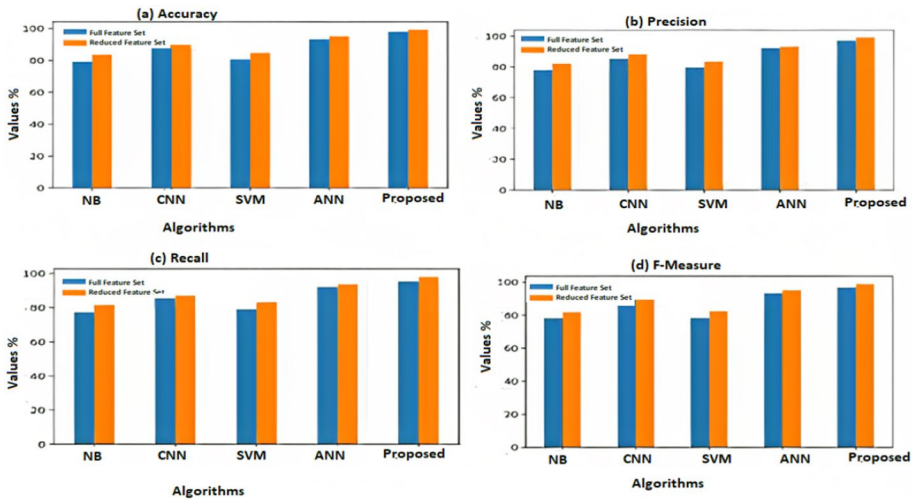

Fig. 2 Comparison between the proposed approach and traditional methods for the NSL-KDD dataset

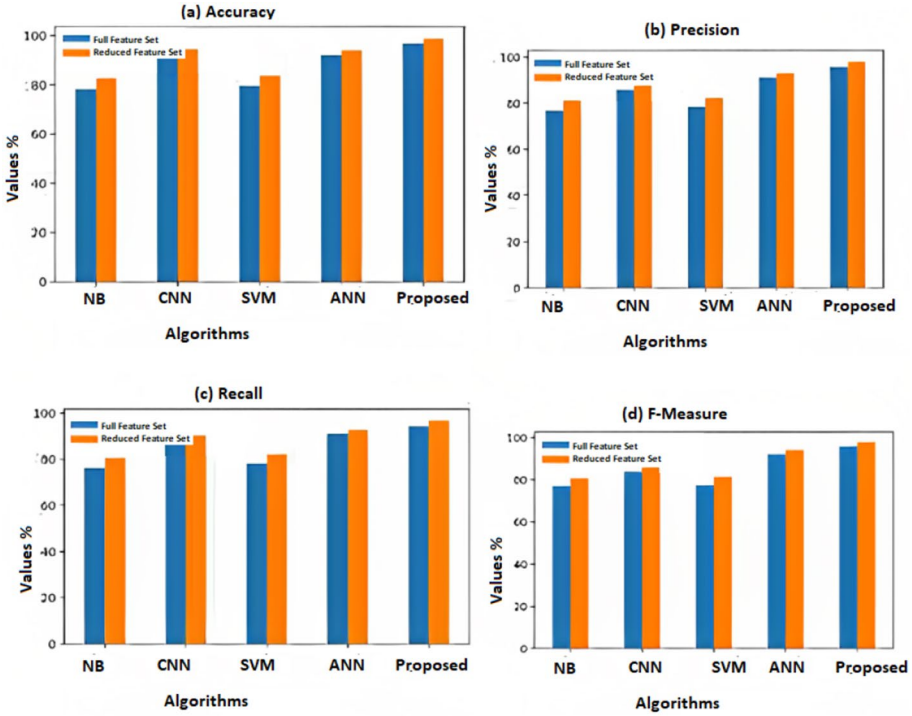

Fig. 3 Comparison between both the proposed and traditional methods for CICIDS2017 dataset

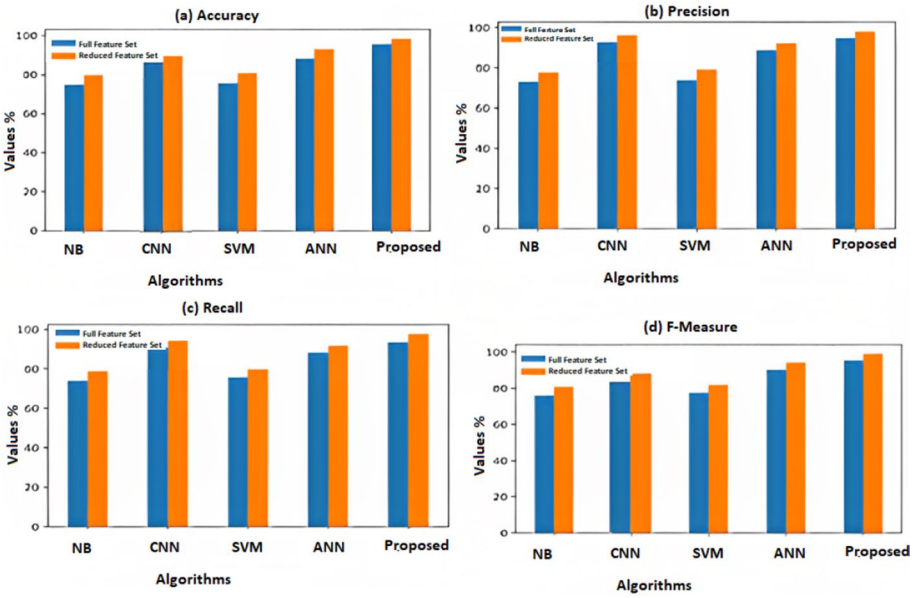

Fig. 4 Comparison between both the proposed and traditional methods for the UNSW-NB15 dataset

Contributions to the field

Summarize the key contributions of the research. How does this work advance the state of the art in intrusion detection for WSNs? Highlight the novelty and significance of the intelligent hybrid cyber-security system proposed.

Limitations and caveats

Acknowledge any limitations or caveats in the study. Discuss factors that could affect the generalizability of the results, such as dataset biases or specific conditions of the experiments.

In conclusion, the results and discussion sections play a pivotal role in elucidating the significance and implications of the research. In this case, the paper showcases an innovative approach to enhance WSN security, supported by strong empirical evidence and comparative analysis. These insights provide a comprehensive understanding of the contributions and potential impact of the proposed algorithm in the field of cyber-physical systems security.

In this paper, an evaluation is made of the suggested approach for binary and multi-class classification. The technique tries to create an intrusion detection model using a deep learning algorithm founded on DLFFNN's tenets. 3 datasets, UNSW-NB15, NSL-KDD, & CICIDS2017, are used in the evaluation. To extract and select features, K-means clustering with information gain is used as an approach. The results demonstrate that the suggested DLFFNN-KMC-IG approach outperforms traditional machine learning algorithms in terms of maximum accuracy (ACY), Recall (RE), Precision (PRE), and F1-Score. Moreover, it is observed that DLFFNN models have the ability to recognize more complicated shapes and expose sample occurrences with concealed attributes more precisely than traditional machine learning algorithms. By utilizing the KMC-IG feature reduction approach, the effectiveness of present machine learning classifiers is enhanced. This method outperforms other conventional machine algorithms globally based on the metrics utilized in the study. To perform multi-class classification on the NSL-KDD dataset, the model undergoes training, validation, and testing. In addition, the study discusses the use of deep neural networks (DNNs) as the technique of choice for handling challenging issues in various applications. The data at the input of an artificial neural network (ANN) is determined and transmitted, and each ANN uses an activation function to increase approximating and comprehensibility for every output. While standard machine learning techniques are implemented using the MLib based on Apache Spark, the suggested deep learning model is implemented using the Keras9 package.

The proposed method outperforms existing algorithms like Support Vector Machines (SVM), Naive Bayes (NB), Convolutional Neural Networks (CNN), and Artificial Neural Networks (ANN). However, there are some general reasons why a novel method might show better results compared to traditional algorithms:

1. Feature representation:
   - The proposed method may employ a more effective feature representation or extraction technique compared to traditional algorithms. If the features used by the

model better capture the underlying patterns in the data, it can lead to improved performance.

2. Complexity and non-linearity:
   - Neural networks, including CNNs and ANNs, are capable of capturing complex and non-linear relationships in data. If the problem at hand involves intricate patterns or dependencies, a deep learning approach may have an advantage over linear models like SVM and Naive Bayes.
3. Data imbalances:
   - Traditional algorithms, including SVM and NB, may struggle with imbalanced datasets. If the dataset used for evaluation is imbalanced, the proposed method might incorporate techniques to handle this imbalance, giving it an edge in performance.
4. Hybrid approaches:
   - The proposed method could be a hybrid model that combines the strengths of multiple algorithms. Hybrid models are designed to leverage the advantages of different techniques, potentially resulting in improved performance over individual models.
5. Synthetic data generation:
   - If the proposed method employs techniques like Synthetic Minority Over-sampling Technique (SMOTE) or other data augmentation methods, it can enhance the model's ability to generalize and detect minority classes, which may be challenging for traditional algorithms.
6. Architecture design:
   - The architecture of the proposed model, especially in the case of CNNs or ANNs, might be designed to capture specific domain knowledge or features that are critical for intrusion detection. This tailored architecture can contribute to better performance.
7. Ensemble methods:
   - The proposed method could use ensemble learning, combining multiple models to make predictions. Ensemble methods often lead to more robust and accurate results compared to individual models.
8. Adaptability to domain-specific features:
   - If the proposed method is designed with a deep understanding of the domain and specific characteristics of intrusion detection, it may be better suited to handle the nuances of the problem compared to more generic algorithms.

It's important to note that the effectiveness of a method depends on various factors, including the dataset, problem complexity, and the specific design choices made in each algorithm. Without detailed information on the proposed method's architecture, features, and evaluation metrics, it's challenging to provide a more specific explanation for its superior performance over SVM and NB in the context of intrusion detection.

Conclusion

This study focused on three key datasets: NSL-KDD, UNSW-NB 15, and CICIDS 2017, and evaluated the accuracy, precision, recall, and F-measure of the proposed approach under two different scenarios: full features and reduced features. The proposed

DLFFNN-KMC-IG was also compared to benchmark machine learning approaches. In the reduced feature set, the proposed algorithm achieved an accuracy, precision, recall, and F-measure of 99.7%, 99.8%, 97.8%, and 98.8% respectively for the NSL-KDD dataset. The proposed algorithm's accuracy, precision, recall, and F-measure for the CICIDS2017 dataset were 99.8%, 98.7%, 97.7%, and 98.7%, respectively. For the UNSW-NB15 dataset, the proposed algorithm achieved an accuracy, precision, recall, and F-measure of 99.1%, 98.7%, 98.4%, and 99.6% respectively. The study also outlined the settings for the proposed hybrid system with feature reduction for machine learning for attack classification and the parameters for the generic machine-learning model. The proposed intelligent hybrid cyber-security system was crucial for recognizing and preventing related attacks in WSN environments. The system effectively reduced the features of the dataset for classification using ML SVD and PCA, and by combining ML and DL, the system provided high-performance features for efficient early detection and learning systems.

Acknowledgements
The authors would like to thank the editorial and review committees for the upcoming review and for the valuable time that will be devoted to the review.

Author contributions
The researchers cooperated together in proposing the idea of research, implementation, and conducting the research. The research was written by the first researcher, Mohamed Behairy, and the second researcher, Mohamed Ali, reviewed.

Funding
Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Availability of data and materials
All data generated or analyzed during this study are included in this published article [and its supplementary information files].

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 22 May 2023 Accepted: 14 December 2023
Published online: 13 January 2024

References

1. Saber AM, Behiry MH, Amin M. Real-time optimization for an AVR system using enhanced Harris Hawk and IIoT. Stud Inform Control. 2022;31(2):81–94.
2. Behiry MH, Amin M, Sauber AM. IIoT-based automatic FOPID tuning for AVR systems using a customized chaotic whale optimization. https://www.doi-i.org/journals/view/373.
3. Saini GK, Halgamuge MN, Sharma P, Purkis JS. A review on cyberattacks. In: Shaikh RA, editor. Secure cyber-physical systems for smart cities. Pennsylvania: IGI Global; 2019. p. 183–219. https://doi.org/10.4018/978-1-5225-7189-6.ch008.
4. Chelli K. Security issues in wireless sensor networks: attacks and countermeasures. Proceedings of the World Congress on Engineering, Vol I, London, UK. 2015.
5. He D, Chan S, Guizani M. Cyber security analysis and protection of wireless sensor networks for smart grid monitoring. IEEE Wirel Commun. 2017;24(6):98–103. https://doi.org/10.1109/MWC.2017.1600283WC.
6. Padmavathi G, Shanmugapriya D. A survey of attacks, security mechanisms and challenges in wireless sensor networks. (Cornell University). 2009. https://arxiv.org/pdf/0909.0576.
7. Pathan AK, Lee H-W, Hong CS. Security in wireless sensor networks: issues and challenges. Proc. ICACT 2006; 1, 20–22: 1043–1048.
8. Perrig A, Stankovic J, Wagner D. Security in wireless sensor networks. Commun ACM. 2004;47(6):53–7.
9. Jian-hua LI. Cyber security meets artificial intelligence: a survey. Front Inf Technol Electron Eng. 2018;19:1462–74.

10. Handa A, Sharma A, Shukla SK. Machine learning in cybersecurity: a review. WIREs Data Mining Knowl Discov. 2019. https://doi.org/10.1002/widm.1306.
11. Thomas T, Vijayaraghavan AP, Emmanuel S. Machine learning approaches in cyber security analytics. eBook, Springer Nature Singapore 2020.
12. Saini GK, Halgamuge MN, Sharma P, Purkis JS. A review on cyberattacks: security threats and solution techniques for different applications. In: Shaikh RA, editor. Secure cyber-physical systems for smart cities. Pennyslvania: IGI Global; 2019. p. 183–219. https://doi.org/10.4018/978-1-5225-7189-6.ch008.
13. Boussi GO, Gupta H. A proposed framework for controlling cyber-crime. 8th International Conference on Reliability. Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) 2020; pp.1060–1063.
14. Kumar G, Kumar K, Sachdeva M. The use of artificial intelligence-based techniques for intrusion detection: a review. Artif Intell Rev. 2010;34(4):369–87.
15. Johri P, Verma JK, Paul S, editors. Applications of machine learning. Singapore: Springer Singapore; 2020.
16. Saleem S, Ullah S, Yoo HS. On the security issues in wireless body area networks. Int J Digit Content Technol Appl. 2009. https://doi.org/10.4156/jdcta.vol3.issue3.22.
17. Sharma K, Ghose MK. Wireless sensor networks: an overview on its security threats. IJCA Special Issue on Mobile Adhoc Networks 2010.
18. Martins D, Guyennet H. Wireless sensor network attacks and security mechanisms: a short survey 2010; IEEE.
19. Sastry AS, Sulthana S, Vagdevi S. Security threats in wireless sensor networks in each layer. Int J Adv Netw Appl. 2013;4(4):1657–61.
20. Kaplantzis S. Security models for wireless sensor networks 2006. http://members.iinet.com.au/~souvla/transferfinal-rev.pdf.
21. Karlof C, Wagner D. Secure routing in wireless sensor networks: attacks and countermeasures. Ad Hoc Netw J. 2003;1(2–3):293–315.
22. Yu Y, Li K, Zhou W, Li P. Trust mechanisms in wireless sensor networks: attack analysis and countermeasures. J Netw Comput Appl. Elsevier, 2011.
23. Xu W, et al. The feasibility of launching and detecting jamming attacks in wireless networks. MobiHoc '05: Proc. 6th ACM Int. Symp. Mobile Ad Hoc Net. and Comp 2005; pp. 46–57.
24. Xu W, Trappe W, Zhang Y. Channel surfing: defending wireless sensor networks from interference. In Proc. of Information Processing in Sensor Networks 2007.
25. Sohrabi K, Gao J, Ailawadhi V, Pottie GJ. Protocols for self-organization of a wireless sensor network. IEEE Pers Commun. 2000;7:16–27.
26. Raymond DR, Midkiff SF. Denial-of-service in wireless sensor networks: attacks and defenses. IEEE Pervasive Comput. 2008;7(1):74–81.
27. Parno B, Perrig A, Gligor V. distributed detection of node replication attacks in sensor networks. Proceedings of the IEEE Symposium on Security and Privacy (S&P'05) 2005.
28. Xiao Y, Rayi VK, Sun B, Du X, Hu F, Galloway M. A survey of key management schemes in wireless sensor networks. Comput Commun. 2007;30(11–12):2314–41.
29. Jain A, Kant K, Tripathy MR. Security solutions for wireless sensor networks. In: Second international conference on advanced computing & communication technologies. 2012. https://doi.org/10.1109/acct.2012.102.
30. Burgner DE, Luay A. Wahsheh security of wireless sensor networks. In: Eighth international conference on information technology: new generations. 2011, pp. 315–20. https://doi.org/10.1109/ITNG.2011.62.
31. Zhu S, Setia S, Jajodia S. LEAP: efficient security mechanisms for large-scale distributed sensor networks. Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03) 2003; 62–72
32. Culler DE, Hong W. Wireless sensor networks. Commun ACM. 2004;47(6):30–3.
33. Makhija J, Appu Shetty A, Bangera A. Classification of attacks on MQTT-Based IoT system using machine learning techniques. Part of the Advances in Intelligent Systems and Computing book series 2021; vol. 1394, 29.
34. Ma T, Wang F, Cheng J, Yu Y, Chen X. A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks. Sensors. 2016;16(10):1701.
35. Zhang F, Chan PP, Biggio B, Yeung DS, Roli F. Adversarial feature selection against evasion attacks. IEEE Trans Cybern. 2015;46(3):766–77.
36. Tavallaee M, Bagheri E, Lu W, Ghorbani AA. A detailed analysis of the KDD CUP 99 data set. In IEEE symposium on computational intelligence for security and defense applications 2009; 1–6.
37. Sonule AR, Kalla M, Jain A, Chouhan DS. UNSWNB15 dataset and machine learning based intrusion detection systems. Int J Eng Adv Technol (IJEAT). 2020;9(3):2638–48.
38. Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp. 2018;1:108–16.
39. Aly M, Alotaibi AS. Molecular property prediction of modified gedunin using machine learning. Molecules. 2023;28:1125. https://doi.org/10.3390/molecules28031125.
40. Johri P, Verma JK, Paul S. Applications of machine learning. Algorithms for Intelligent Systems. eBook, Springer, Nature Singapore. 2020. https://doi.org/10.1007/978-981-15-3357-0
41. AlRikabi HT, Hazim HT. Enhanced data security of communication system using combined encryption and steganography. Int J Interact Mobile Technol. 2021. https://doi.org/10.3991/ijim.v15i16.24557.
42. Ahmad R, Wazirali R, Abu-Ain T. Machine learning for wireless sensor networks security. An overview of challenges and issues. Sensors. 2022;22:4730.
43. Ismail S, Khoei TT, Marsh R, Kaabouch N. A comparative study of machine learning models for cyber-attacks detection in wireless sensor networks. In Proceedings of the 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 1–4 December 2021; pp. 313–318.
44. Khoei TT, Ismail S, Kaabouch N. Dynamic selection techniques for detecting GPS spoofing attacks on UAVs. Sensors. 2022;22:662.
45. Karatas G. Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset. IEEE Access. 2020;8:32150–62.

46. Dong L, Li Y, Gao W. A survey on wireless sensor network security: attacks and defenses. IEEE Access. 2020;8:14237–58.
47. Zhang R, Zhang C, Zhang C. A comprehensive review of wireless sensor network security: real attacks, existing protocols, and open research issues. IEEE Access. 2021;9:12461–86.
48. Li X, Wu D, Zhang X. Machine learning in wireless sensor networks: algorithms, applications, and challenges. Futur Gener Comput Syst. 2022;128:131–48.
49. Wang Z, Ma Y, Jiang B. An intrusion detection system based on deep learning for wireless sensor networks. IEEE Internet Things J. 2021;8(15):12194–203.
50. Chen J, Zhang S, Ma Y. Explainable deep learning for intrusion detection in wireless sensor networks. Ad Hoc Netw. 2023;128: 102933.
51. Kim J, Park S. Clustering-based anomaly detection for wireless sensor networks. Inf Sci. 2020;507:54–66.
52. Jingjing Z, Tongyu Y, Zhang J, Zhang G, Li X, Peng X. Intrusion detection model for wireless sensor networks based on MC-GRU. Wirel Commun Mob Comput. 2022;2022:1–11. https://doi.org/10.1155/2022/2448010.
53. Zhao Y, Li Y, Zhang L. Benchmarking intrusion detection systems in wireless sensor networks: a comprehensive review. Ad Hoc Netw. 2023;128: 102917.
54. Liu Z, Zhang Y, Zhang Y. Trade-off between accuracy and resource consumption in intrusion detection systems for wireless sensor networks. IEEE Internet Things J. 2021;8(24):19589–600.
55. Aly M, Alotaibi NS. A new model to detect COVID-19 coughing and breathing sound symptoms classification from CQT and Mel spectrogram image representation using deep learning. Int J Adv Comput Sci Appl. 2022. https://doi.org/10.14569/IJACSA.2022.0130869.
56. Aly M, Alotaibi NS. A novel deep learning model to detect COVID-19 based on wavelet features extracted from Mel-scale spectrogram of patients cough and breathing sounds. Inform Med Unlocked. 2022;32: 101049.

Publisher's Note