IOT SECURITY ENHANCEMENT VIA DECISION TREE RFE AND PEARSON CORRELATION

Dr. S.M.N. Yusuf

ABSTRACT

The recent growth on the Internet of Things (IoT) has been a major driver of the industrial revolution, especially with the rise of smart cities. IoT applications automate real-time processes, making life more efficient and convenient. These IoT-enabled systems enhance the quality of life, improve service quality, and increase operational efficiency. However, this technology also poses significant security risks. As smart devices handle sensitive data, there is a growing risk of cyber-attacks, data misuse, and privacy breaches, which can undermine user trust and hinder the progress of smart city development. To address these cybersecurity challenges, it's essential to develop a model that can protect IoT devices from various threats in real time. This paper proposes an innovative approach that combines efficient feature selection and fusion techniques to enhance intrusion detection in IoT systems. The proposed model starts by preprocessing the raw IoT data, preparing it for the next stages of analysis. The Decision Tree-based Pearson Correlation Recursive Feature Elimination (DT-PCRFE) method is then applied to select the most relevant features. By eliminating redundant and uncorrelated data, this step not only improves resource utilization but also reduces computational complexity, making the system faster and more efficient. After selecting key features, the model employs a feature fusion process, converting IoT device requests into word embeddings. This transformation strengthens the model's ability to detect diverse attack patterns, thus increasing its robustness.

***Index Terms**: Attack Detection, Internet of Things (IoT), Deep learning, Decision Tree, Recursive Feature.*

Reference to this paper should be made as follows:

Dr. S.M.N. Yusuf, (2025), "IoT Security Enhancement Via Decision Tree RFE and Pearson Correlation" Int. J. of Electronics Engineering and Applications, Vol. 13, No. 1, pp. 01-19.

Biographical notes:

***Dr. S.M.N Yusuf** is currently a Senior Lecturer in the Department of Physics, Nasarawa State University, Keffi, Nigeria. He holds BSc. in Physics in 1999 and MSc. in applied Physics in 2010 from University of Jos, Nigeria. He obtained his Ph.D. in Engineering in 2005 from Seoul National University, South Korea. He has over 65 publications in reputed journals and conferences. His areas of interest include Electronics Circuit Design and Analysis, Telecommunication, and Medical Physics. He is a Member of Nigerian Institute of Physics.*

1. INTRODUCTION

The recent surge in technology advancements is increasingly centered on automating various processes through interconnected devices and computer networks. This shift is revolutionizing industries and improving quality of life by enabling seamless data exchanges across IoT-enabled systems. IoT sensors generate vast amounts of data that are processed and transmitted across multiple fields, including healthcare, retail, transportation, and automotive industries. This continuous data flow supports automation, optimizes operations, and enhances service delivery. As industries explore IoT's potential, they have found that integrating Machine Learning (ML) and Deep Learning (DL) models significantly boosts efficiency, productivity, and reliability. These models leverage real-time data collected through IoT sensors, applications, and programs to optimize decision-making, streamline operations, and meet customer needs. Consequently, businesses across sectors are seeing improvements in goods and services, ethics, and organizational adaptability. In IoT systems, machine-to-machine (M2M) and person-to-person communications rely on network packets and protocols. However, these communication mechanisms often contain vulnerabilities that attackers can exploit. Flaws in protocols make IoT devices and networks susceptible to data breaches and other cyber-attacks. Without strong cybersecurity measures, network attackers could disrupt IoT systems, with potential economic damages estimated to reach $90 trillion by 2030. Among the greatest risks to IoT devices are malware and zero-day vulnerabilities. Attackers often use techniques like Denial of Service (DoS), Progressive Determined Risks (PR), and Decentralized DoS (DDoS) to target IoT systems, threatening the stability of critical infrastructure. While current security measures—such as security protocols, access control mechanisms, biometric authentication, and cryptography—help mitigate risks, they are often insufficient to fully protect IoT systems from advanced threats.

Given the rapid evolution of cyber threats, advanced attack detection systems have become essential. These systems apply sophisticated algorithms to identify and address vulnerabilities in IoT networks. Research indicates that around 70% of IoT devices face significant security risks, with as many as 15 types of vulnerabilities, including encryption and password security gaps. These weaknesses make IoT devices attractive targets for attacks, particularly in applications like smart homes, transportation, agriculture, healthcare, smart grids, and earthquake detection systems. Malicious actors exploit IoT vulnerabilities, turning devices into potential attack vectors across different domains. IoT devices frequently use wireless communication, increasing exposure to cyber threats as wireless networks are easier for attackers to access. Unlike conventional network threats that may be confined to specific areas, IoT-based attacks have a wide reach, making their impact particularly dangerous and complex to control. To protect IoT networks from these cyber threats, robust security measures are crucial. IoT security must evolve rapidly, integrating multi-layered defenses, real-time monitoring, and enhanced encryption to ensure a safe and resilient network infrastructure. Strengthening IoT cybersecurity will be key to mitigating vulnerabilities and establishing secure IoT frameworks capable of withstanding emerging threats. As IoT applications expand, adaptive and resilient security strategies are vital for safeguarding critical infrastructure. By advancing IoT cybersecurity, industries can capitalize on the benefits of automation without compromising safety, ensuring that IoT systems remain reliable foundations for the future of digital transformation. With comprehensive detection and prevention mechanisms in place, organizations can manage risks effectively, allowing IoT to drive innovation and productivity safely across sectors.

1.1. Objectives

- **Review of Recent Research**: Examining current papers on IoT attack detection models highlights key challenges in achieving accurate threat identification amidst varied data patterns and evolving attack types.

- **Efficient Feature Selection**: Identifying relevant features is crucial for boosting detection accuracy while reducing computational demands, emphasizing the need for advanced feature selection models.
- **Optimal Hyperparameter Tuning**: Selecting the best hyperparameters for classification models is essential to minimize false predictions, improving detection reliability.
- **Goal**: These strategies collectively aim to create robust IoT security frameworks, balancing high detection accuracy with efficient resource use.

1.2. Motivation

IoT technology is rapidly transforming sectors such as home automation, healthcare, smart cities, and manufacturing, pushing both public and private sectors toward connected, knowledge-based networks. As IoT applications expand in areas like smart homes, transportation, and power grids, creating effective policies and deploying advanced IoT solutions have become increasingly complex. Anomaly detection in IoT systems is now a prominent research area due to the rising frequency of cyber threats across all fields where IoT is implemented. The layered structure of IoT protocols allows for constant introduction of new attacks, often slight variations of known threats, challenging traditional security measures like cryptography, which may not detect minor modifications in time. Notably, the success of Machine Learning (ML) and Deep Learning (DL) in various data-driven sectors has inspired their use in IoT cybersecurity, where they enhance attack detection and classification. This approach emphasizes using ML and DL for efficient feature selection and classification to boost detection accuracy in IoT systems. By applying these advanced techniques, IoT environments can be safeguarded against evolving cyber threats, ensuring secure, resilient infrastructure across essential sectors.

1.3. Contribution of the work

This paper contributes to enhancing IoT attack detection through a multi-stage model that incorporates data preprocessing, feature selection, feature fusion, and classification for improved accuracy and efficiency.

- **Preprocessing**: The input data is preprocessed using four methods to improve dataset quality:
    1. **Data Cleaning** to remove errors and inconsistencies,
    2. **Log Processing** for capturing detailed data insights,
    3. **Normalization** to ensure data consistency, and
    4. **One-Hot Encoding** to convert categorical data for machine learning models. Together, these methods prepare the BoT-IoT dataset for accurate training and reduce noise for the model.
- **Feature Selection**: An enhanced feature selection approach using Decision Trees (DT) with Pearson Correlation-based Recursive Feature Elimination (DT-RFE) is proposed. DT-RFE efficiently identifies the most relevant features, reducing feature dimensions by eliminating redundant and uncorrelated data, thus boosting model efficiency and accuracy.
- **Feature Fusion**: The proposed model employs a Multimodal Feature Fusion approach that assigns weight values to selected features, enhancing robustness and interpretability. These weights prioritize impactful features, which are then fed into the neural network, optimizing the model's focus during the classification phase.

- **Classification**: An optimized Deep Neural Network (DNN) is used to classify and predict attack types based on the weighted, selected features from DT-RFE. The DNN's optimized structure enables high accuracy in detecting a variety of IoT threats.

The paper's structure includes:

- **Section 2**: A discussion of related work in IoT security and feature selection.
- **Section 3**: An overview of the dataset, detailing feature descriptions used for model training.
- **Section 4**: Introduction to the proposed system model and methods from preprocessing to classification.
- **Section 5**: Experimental setup and comparative analysis of results, demonstrating the model's effectiveness over existing methods.
- **Section 6**: Conclusion, summarizing the model's contributions and suggesting future research directions in IoT security.

In summary, this paper presents a robust framework that balances detection accuracy, interpretability, and computational efficiency, advancing secure IoT environments.

2. RELATED WORKS

This section reviews recent studies on IoT attack detection models, focusing on methods that strengthen network security. Kan et al. [10] highlighted the importance of identifying attacks in IoT networks and proposed an Adaptive Particle Swarm Optimization-Convolutional Neural Network (APSO-CNN) approach. By using particle swarm optimization, their method fine-tunes one-dimensional CNN parameters to enhance detection performance, with cross-entropy loss of the trained CNN as the fitness metric, demonstrating effective threat detection. Parthasarathi et al. [23] focused on a decision tree-based model for managing keys in IoT group communication. Their approach uses a decision tree structure to manage keys efficiently, thereby improving security in IoT networks. Simulation results confirmed the model's reliability and effectiveness in supporting IoT attack detection. Pecori et al. [24] examined IoT's role in daily life, particularly in network traffic detection and classification. They introduced a comprehensive dataset designed for monitoring network traffic and applied deep neural networks to classify traffic into binary and multi-class categories. Their model effectively identifies and categorizes network traffic patterns, addressing IoT-specific challenges in monitoring and threat detection. Together, these studies highlight the strength of machine learning and deep learning techniques, such as APSO-CNN, decision trees, and deep networks, in advancing IoT network security by enabling efficient threat detection and traffic classification in complex IoT environments. Nimbalkar and Kshirsagar [22] investigated various IoT attacks arising from device vulnerabilities and the challenges of using machine learning (ML) for detection due to the complex features of IoT traffic. They introduced a feature selection method for intrusion detection systems to detect Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. Their approach applies insertion and union operations to create feature subsets for the proposed model. Using the IoT-BoT and KDD Cup 1999 datasets with a JRip classifier, they validated the method's effectiveness in selecting critical features to improve detection.

Atul et al. [5] emphasized the need for secure digital communication in IoT but noted that security challenges like anomalies and service failures remain prevalent. They proposed the Energy-Aware Smart Home (EASH) framework, a communication model for smart home networks aimed at detecting anomalies and strengthening security. Their model uses machine learning to identify irregular

communication patterns, distinguishing between security threats and service-related issues. EASH was evaluated for performance, accuracy, and efficiency, demonstrating its capacity to enhance secure and reliable communication within smart homes. Rahman et al. [27] developed an Intrusion Detection System (IDS) for smart cities, titled Scalable Machine Learning for IoT-Enabled Smart Cities. They addressed the limitations of centralized IDS by proposing a semi-distributed and distributed model, specifically designed for the complex environments of smart cities. The system incorporates advanced feature extraction and selection techniques to handle large-scale IoT data. By leveraging parallel machine learning, their approach allocates tasks efficiently across networks, yielding high detection accuracy and faster model-building times, making it a robust option for securing IoT networks in smart cities. Gu et al. [7] explored security and privacy issues in IoT, noting the increasing threat to IoT networks and its implications for human safety. They proposed a reinforcement learning-based model for threat detection, capable of recognizing and adapting to changing attack patterns. Their model also analyzed IoT traffic characteristics, employing entropy-based metrics to predict threats based on changes in traffic patterns, providing an effective defense against evolving security risks. These studies collectively emphasize the potential of advanced ML and distributed systems for enhancing IoT security. Through methods such as feature selection, communication frameworks, scalable IDS, and reinforcement learning, these approaches address unique challenges across IoT applications like smart homes and cities. By responding to the specific demands of IoT environments, these works significantly contribute to the development of resilient and secure IoT systems.

Krishna and Thangavelu [12] addressed the challenge of Denial of Service (DoS) attacks in IoT systems, exploring security issues unique to IoT devices. To improve detection, they proposed a hybrid meta-heuristic algorithm, combining the Lion Optimization Algorithm (LOA) and the Firefly Algorithm (FFA), named ML-F. This approach aimed to enhance accuracy in identifying attacks. By applying their model to NSL-KDD and IoT datasets, they achieved high performance and effective attack classification. Lian et al. [31] developed an advanced detection approach by combining Decision Tree and Recursive Feature Elimination (RFE) for feature selection. They used a stacking fusion model, integrating multiple machine learning algorithms to optimize detection outcomes. Testing on the NSL-KDD dataset, their approach achieved over 98% accuracy, demonstrating the efficiency of stacking models in improving the reliability of IoT security systems. Yang et al. [32] introduced an Intrusion Detection System (IDS) that leverages both knowledge graphs and statistical feature selection techniques. Their model utilized Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) networks to identify malicious activities. With an accuracy of 90.01% on the NSL-KDD dataset, their model underscored the potential of using deep learning architectures to detect IoT-based threats effectively. Recent developments in IoT attack detection emphasize combining deep learning, optimization algorithms, and machine learning to address evolving security threats. Sagu et al. [26] introduced a hybrid Neural Network (NN) model that integrates Convolutional Neural Networks (CNN) with Deep Belief Networks (DBN) for effective attack detection. They enhanced this model using the Seagull Adopted Elephant Herding Optimization (SAEHO) technique, which adjusts the weights to improve detection precision.

Anwer et al. [4] evaluated machine learning approaches for detecting malicious IoT traffic, specifically testing Support Vector Machines (SVM), Gradient Boosted Decision Trees (GBDT), and Random Forest (RF). The RF model performed best, achieving an accuracy of 85.34% on the NSL-KDD dataset, underscoring its effectiveness in attack detection. Inayat et al. [8] conducted a review of learning-based methods in machine learning (ML) and deep learning (DL) for IoT attack detection, analyzing recent advancements and identifying future research directions. They highlighted the need for models that can quickly adapt to the changing threat landscape in IoT. Yadav et al. [30] designed an Autoencoder (AE) with Deep Neural Networks (DNN) to improve the speed and accuracy of attack detection in 5G

networks. This model reduced detection time and significantly improved accuracy, achieving a 99.76% success rate, demonstrating the potential of combining AE and DNN for advanced networks. Garagei et al. [1] examined a variety of machine and deep learning models, including Decision Tree, SVM, K-Nearest Neighbors (KNN), Ensemble Learning (EL), PCA, CNN, AE, RNN, and GAN, applied across multiple datasets to improve IoT system detection accuracy. Ioannou et al. [9] focused on detecting specific network attacks like forward and blackhole attacks, using SVM and testing on the IoT test bed dataset. Despite these advances, existing models still struggle to reach the desired levels of accuracy and robustness. Enhancing feature selection methods could increase the reliability of IoT attack detection. In this paper, a new Feature Re-selection model is introduced, incorporating a deep learning classifier to further boost accuracy and resilience against attacks in IoT environments.

3. MATERIALS AND METHODS

The BoT-IoT dataset, created by the Cyber Center at the University of New South Wales in 2018, provides a realistic representation of IoT network traffic, encompassing both normal and malicious data. This dataset captures typical traffic patterns in IoT networks with attacks simulated by intelligent devices such as smart fridges, motion-controlled lights, remotely operated garage doors, and intelligent thermostats. With approximately 73 million instances and 42 features, each data point is labeled as either an attack or normal traffic. Attacks are categorized into four primary types: Denial of Service (DoS), Distributed Denial of Service (DDoS), reconnaissance, and data theft (intelligence stealing). The dataset has been preprocessed to remove redundant features, enhancing its usability for machine learning applications. The remaining essential features and types of attacks are detailed in Tables 1 and 2, making this dataset a valuable resource for developing and testing IoT security models focused on attack detection.

Table 1. BoT-IoT superfluous feature set

Feature Number	Feature	Description	Data type
f1	_pkSequence ID	Row_identifier	Int
f2	_Start time	Record_Start_Time	Float
f3	_Flags	Flow state flag	Categorical
f4	_Protocol	Protocol textual representation	Categorical
f5	_Source port	Port number of the source	Categorical
f6	_Destination port	Port number of the destination	Categorical
f7	_Packets	Total number of packets in a transaction	Int
f8	_State	State of the transaction	Categorical
f9	_Dur	The total duration of the record	Float
f10	_Mean	Aggregated record avg duration	Float
f11	_Dpkts	Count of destination to source	Int

f12	_Sbytes	Byte count from source to destination	Int
f13	_Dbytes	Byte count from destination to source	Int
f14	_TBpSIP	Total No. of bytes per source IP	Int
f15	_TPPProto	Total No. of packets per protocol	Int
f16	_ARPProto PSrcIP	Avg rate per protocol per source IP	Float

Table 2. Instances of attacks using the BoT-IoT dataset

BoT-IoT	Category	Total number of instances
Normal		9656
Attacks	DoS	254653
	DdoS	12123563
	Reconnaissance	1593446
	Information theft	1657

4. PROPOSED METHODOLOGY DESIGN

This section outlines the architecture of the proposed IoT attack detection model, detailing each phase: data preprocessing, feature selection, and detection. Preprocessing involves data cleaning and transformation to improve quality, followed by feature selection to reduce dimensionality and retain relevant features. The detection phase then applies advanced methodologies for accurate attack identification.

4.1 System Architecture

The proposed attack detection model's architecture is illustrated in Figure 1 and comprises four essential phases: (i) preprocessing, (ii) feature selection, (iii) feature fusion, and (iv) classification for attack detection.

In the **preprocessing phase**, the quality of the dataset is enhanced using various techniques. Data cleaning removes inconsistencies, log processing organizes and formats the data, normalization standardizes the data range, and one-hot encoding transforms categorical variables into numerical values.

The **feature selection phase** employs the Decision Tree-based Pearson Correlation Recursive Feature Elimination (DT-PCRFE) approach to identify and extract the most relevant features from the dataset. This method effectively reduces dimensionality by discarding redundant and irrelevant data.

During the **feature fusion phase**, the selected features are combined, and weight values are assigned to enhance system robustness. This process improves performance and increases interpretability, clarifying the significance of each feature.

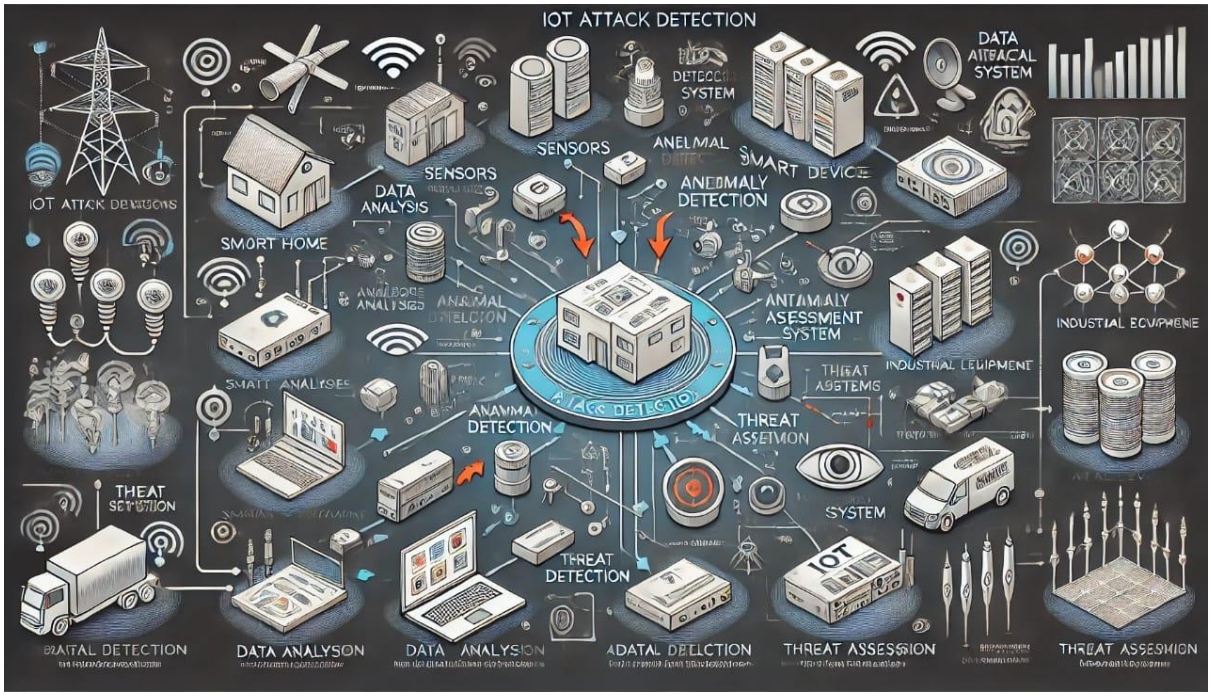

Fig.1. An overview of the suggested IoT attack detection model

Finally, the **classification phase** utilizes a Deep Neural Network (DNN) classifier to detect malicious requests or attacks from IoT devices, leveraging the extracted features to ensure optimal performance in identifying threats.

4.2. Data Preprocessing

Processing the initial input data for attack detection in IoT networks is challenging due to the extensive volume of network traffic, which consists of various features in both numeric and non-numeric forms. To tackle the issues associated with non-numerical features and enhance dataset quality, this paper implements four preprocessing techniques: data cleaning, log processing, normalization, and one-hot encoding.

Data cleaning is crucial for improving dataset integrity, as it identifies and eliminates redundant data, including duplicate values. This cleaning process is applied to both training and testing datasets to ensure consistency and accuracy. Furthermore, symbolic features are transformed into numerical formats since machine learning (ML) and deep learning (DL) models require real-number vectors for processing. The feature labels are categorized into normal and abnormal network traffic, with the latter classified into specific attack types: Denial of Service (DoS), Distributed Denial of Service (DDoS), reconnaissance, and information theft.

Log processing helps to minimize discrepancies among feature values, effectively addressing the variance in the data. Features with significantly larger values are flagged as outliers, which can negatively impact system performance. To mitigate this issue, the log function, as outlined in Equation (1), is applied to normalize feature values, ensuring they share a consistent granularity. This dimensionality reduction not only enhances the overall performance of the model but also ensures that all features contribute equally to the detection process, ultimately improving accuracy in identifying attacks within the IoT environment.

$$D_i' = \log(f_i), \quad (1)$$

In this context, f_i denotes feature i. The process of feature normalization is essential for converting feature values into a consistent range, addressing issues of data imbalance and the tendency for larger values to dominate. Utilizing the Python scikit-learn library, symbols are mapped to unique numeric values through a systematic process. Normalization plays a crucial role in ensuring that all data values fall within the same range, which is vital for optimal feature selection. In this study, min-max normalization is employed to scale feature values to a range between 0 and 1. This method effectively minimizes discrepancies among feature values, fostering a more balanced contribution during subsequent analyses and enhancing the model's overall performance, as expressed in Equation (2).

$$D_{ij}' = \frac{X_{ij} - \min(f_i)}{Max(f_i) - Min(f_i)}, \quad (2)$$

In this process, iii represents a feature, and jjj denotes a record within the dataset, while Max(fi)\text{Max}(f_i)Max(fi) and Min(fi)\text{Min}(f_i)Min(fi) refer to the maximum and minimum values of the respective features. Through min-max normalization, all continuous features are scaled to a range between [0, 1], emphasizing the importance of each feature relative to the others. Following this, one-hot encoding is applied to convert categorical data into unique binary values. In this scheme, the current category is assigned a value of 1, while all other categories are represented as 0. For instance, a DoS label feature would be encoded as [1, 0, 0, 0, 0]. This transformation enhances the deep learning model by generating improved input vectors, allowing the model to process the data more effectively. The transformed datasets are subsequently used as input for the feature selection process, employing the proposed Decision Tree with the Pearson Correlation Recursive Feature Elimination (RFE) model.

Algorithm 1: Preprocessing

Input: Dataset DDD with features f1,f2,…,fnf_1, f_2, \dots, f_nf1,f2,…,fn

Output: Transformed dataset D′D'D′ with features f1′,f2′,…,fn′f'_1, f'_2, \dots, f'_nf1′,f2′,…,fn′

1. **Initialize**: For each feature fif_ifi in dataset DDD (where i=1i = 1i=1 to nnn):
2. **Check for Symbolic Data**:
    - **If** fif_ifi contains symbolic (non-numeric) values:
        - a. Apply Python's Scikit-learn library to map symbols to numeric values.
3. **Apply Preprocessing Techniques**:
    - Perform the following steps for all features fif_ifi:
        - a. **Data Cleaning**: Remove redundant data and detect duplicates.
        - b. **Log Processing**: Use Equation (1) to transform values, reducing dimensionality.
        - c. **Normalization**: Use Equation (2) to scale values between 0 and 1, minimizing data imbalance.
        - d. **One-Hot Encoding**: Convert categorical features into binary vectors.
4. **Repeat** for each feature until all fif_ifi are preprocessed.
5. **End**

4.3. Utilizing the Suggested Decision Tree with PCRFE for Feature Extraction (DT-PCRFE)

Following data preprocessing, the next essential step is reducing the high dimensionality of the dataset, as this can significantly complicate training in ML and DL models. Effective feature selection plays a crucial role in optimizing detection accuracy by ensuring that only the most relevant features are used for training. In this paper, we propose a novel feature selection method called Decision Tree-based Pearson Correlation Recursive Feature Elimination (DT-PCRFE). DT-PCRFE is an iterative method that evaluates each feature's relevance by examining its correlation and impact on the model's accuracy. Through recursive elimination, it builds and tests the model step-by-step, selecting the most relevant features and removing redundant or insignificant ones. The process is repeated until only the most meaningful features remain, reducing the dataset to a manageable vector for further processing. When a feature shows a nonlinear relationship with the response variable, tree-based methods—such as Decision Trees, which excel at handling nonlinearities without complex debugging—are used. In this model, we employ a Decision Tree for effective feature selection. The Decision Tree (DT) uses an information entropy index to prioritize features. Information entropy quantifies the level of uncertainty or "randomness" in a feature, guiding the tree in determining optimal data splits. The DT recursively calculates entropy, splitting the data layer by layer until each instance is clearly classified. For instance, if D is a variable representing a set of possible outcomes, the probability distribution for D can be described as in Equation (3), representing entropy. Using this entropy measure, the Decision Tree identifies features that provide clearer, more distinct classifications, while those with higher entropy (indicating greater uncertainty) are considered less important and are discarded. By selecting only the most informative features, DT-PCRFE reduces computational complexity, enhances model interpretability, and improves classification performance for attack detection in IoT systems.

$$P(D = f_i) = p_i, \tag{3}$$

The entropy for a random variable D, where each feature f_i corresponds to a probability p_i, is determined using Equation (4).

$$H(D) = -\sum_{i=1}^{n} p_i \log p_i. \tag{4}$$

When entropy H is high, the uncertainty or ambiguity of the random variable D is also elevated, signifying difficulty in predicting its value accurately. This high entropy indicates that the probability p_i for any specific value is below 1, which results in a negative value when taking its logarithm. To handle this, a negative sign is included in the entropy formula, making the output positive. Consequently, a greater probability difference p_i for feature f_i leads to a higher entropy HHH. The joint probability distribution for two random variables, D and G, is expressed in Equation (5).

$$P\left(D = f_i, G = g_j\right) = p_{ij}, \tag{5}$$

In this context, f and g represent the *i-th* and *j-th* features, respectively. The conditional entropy H (G|D) measures the uncertainty of the random variable G given that D is known. This conditional entropy is calculated using Equation (6).

$$H(G|D) = \sum_{i=1}^{n} p_i \, H(G|D = f_i). \tag{6}$$

The Information Gain (IG) of a feature A within the training dataset X, represented as IG (X|A), measures the decrease in uncertainty of X when A is known and is calculated according to Equation (7).

$$IG(X|A) = H(X) - H(X|A). \quad (7)$$

Information Gain (IG) measures the reduction in uncertainty about variable *G* when feature *D* is known. However, relying solely on IG (X|A) for dataset partitioning may result in selecting features with a large number of values, which can distort the analysis. Therefore, IG is refined to address this concern, as illustrated in Equation (8).

$$IG_R = \frac{IG(X|A)}{H(X)}. \quad (8)$$

Let |DT| denote the total number of leaf nodes in the decision tree (DT), with each leaf node represented as *t*. The entropy at each leaf node is denoted by H_t, and N_k indicates the number of samples within that node. The loss function *L* of the decision tree incorporates a penalty term $\rho \geq 0$ to mitigate overfitting, as shown in Equation (9).

$$L_\rho(DT) = \sum_{t=1}^{|DT|} N_t H_t(DT) + \rho(DT), \quad (9)$$

Entropy is determined by the probabilities of the class distributions.

$$H_t(DT) = -\sum_k \frac{N_{tk}}{N_t} log \frac{N_{tk}}{N_t} \quad (10)$$

The loss function measures the difference between the actual values and the model's predicted outcomes. The key objective of the learning algorithm is to minimize this loss function to enhance prediction accuracy. Consequently, the first component in Equation (9) focuses on calculating the entropy of the leaf nodes, which indicates the model's effectiveness in classification.

$$R(T) = \sum_{t=1}^{|DT|} N_t H_t(DT) = -\sum_{t=1}^{|DT|} \sum_{k=1}^{K} N_{tk} log \frac{N_{tk}}{N_t}. \quad (11)$$

The loss function can be simplified as:

$$R_\rho(DT) = R(DT) + \rho(DT), \quad (12)$$

In the proposed feature selection model, R(T) indicates the prediction error of the training data, while |DT| signifies the complexity of the decision tree model. The penalty term ρ plays a crucial role in balancing model complexity and prediction accuracy, ensuring that the model avoids excessive complexity that could undermine its performance.

The Pearson Correlation Recursive Feature Elimination (PCRFE) method systematically evaluates different combinations of features by calculating the sum of their respective coefficients. This process assigns a score to each feature, enabling the identification and retention of the most significant ones for further analysis. The Pearson Correlation (PC) quantifies the strength of the linear relationship between two variables, with values ranging from -1 to 1. A value of +1 indicates a perfect positive correlation, 0 denotes no correlation, and -1 reflects a perfect negative correlation. Unlike conventional machine learning feature selection techniques, which often assess features individually, PCRFE efficiently eliminates irrelevant features in groups. The correlation coefficient for features is computed using Equation (13), streamlining the feature selection process and enhancing overall efficiency.

$$PCorr_{fi,gi} = \frac{\sum_{i=1}^{n}(f_i-\bar{D})(g_i-\bar{G})}{\sqrt{\sum_{i=1}^{n}(f_i-\bar{D})^2}\sqrt{\sum_{i=1}^{n}(g_i-\bar{G})^2}}, \quad (13)$$

In the feature selection process, f_i, and g_i denote the features being evaluated for correlation. The correlation coefficient, which ranges from -1 to 1, reflects the strength and direction of the relationship between these features. Values close to -1 or 1 indicate a strong correlation, while a value near 0 suggests a weak relationship. To optimize the feature set, a threshold value is used to rank the correlated features. Features that rank below this threshold are considered less significant and are subsequently eliminated from the dataset. This computation for feature removal is outlined in Equation (14), enhancing the feature selection process by retaining only the most relevant features for further analysis.

$$PCRFE(f_i) = \sum_{i=1}^{n}\left(g_{i,j} - \sum_{j=1}^{d} PCorr_{f_i,g_j} \times f_i\right)^2. \quad (14)$$

In this paper, we present a feature fusion approach within the feature extraction phase to enhance classification accuracy. This multimodal feature fusion method effectively combines features selected from the feature selection phase, assigning weights to each feature to improve performance in the neural network classification process. The step-by-step procedure is outlined in Algorithm 2, starting with the selection of pairs of features as inputs for processing. The decision-making model is then trained to compute a ranking criterion for these feature pairs. Correlation is assessed for all feature pairs in the dataset, and based on a predefined threshold, correlated features are chosen and included in the subset RRR, while non-correlated features are eliminated. For the selected features, a weight set is created by assigning weight parameters to each feature; if a feature is part of the selected subset, its weight is multiplied by an optional parameter. Conversely, the weight remains unchanged for features outside this subset. These weight parameters are essential for the subsequent classification process, ensuring that relevant features effectively contribute to attack detection in the IoT environment.

Algorithm 2: DT-PCRFE Feature Selection and Feature Fusion outlines the following steps for selecting relevant features and performing feature fusion in the context of IoT attack detection:

1. **Initialize Sets:**
    - Set *W* for weights as $W=\emptyset$.
    - Set *R* for the feature order set as $R=\emptyset$.
2. **Train Decision Tree:**
    - For each feature *i* from 1 to *N*:
        - Train the decision tree and calculate the ranking criterion.
3. **Feature Pair Processing:**
    - For each feature pair <*fi,gi*>in set *S*:
        - Compute the correlation coefficient of the features using Equation (13).
        - Remove irrelevant features based on the PCFSR using Equation (14).
4. **Threshold Evaluation:**
    - If the condition PCFSR *(fi)*≥threshold is met:
        - Add feature *fi* to the selected feature subset *R*.

- Otherwise, remove the feature from consideration.

5. **Weight Calculation:**
   - For each feature fi in set S:
     - Calculate the weight using the formula: Weight=$\rho \times w$ *(if* $fi \in R$*)*
     - If fi is not in R, the weight remains unchanged.
6. **Output:**
   - Return the final selected feature set R along with their corresponding weights W.

This structured approach ensures the selection of the most relevant features while assigning appropriate weights to enhance the performance of the model in detecting IoT attacks.

The proposed feature selection model effectively refines the original dataset of 42 features, focusing only on the most relevant ones for improved detection and processing efficiency. During preprocessing, redundant and irrelevant features are filtered out, reducing the dataset to a refined set of 19 essential features, as shown in Table 1. Among these, nine critical features— R={f3, f4, f7, f8, f12, f13, f16, f17 and f18} have been identified for their significant contribution to the detection process. This carefully selected subset captures the key patterns needed for accurate attack detection, allowing the classifier to operate effectively without the noise of superfluous data. The optimized subset is then used as input to the classifier, facilitating precise and efficient IoT attack detection.

4.4 Classification – Optimized DNN

The proposed classification model utilizes a Deep Neural Network (DNN) with optimized hyperparameters tailored for effective IoT attack detection. Key hyperparameters—such as learning rate, epoch size, momentum, batch size, and dropout regularization—are tuned to enhance the model's performance. To achieve the optimal settings for these parameters, a Random Search approach is employed, in which the model undergoes iterative training with randomly selected parameter combinations. This process continues over a predefined number of iterations, systematically refining the model for improved performance. A Deep Neural Network is an advanced form of Artificial Neural Network (ANN) that comprises multiple hidden layers between the input and output layers, allowing it to capture complex patterns and dependencies within the data. In this DNN model, each layer plays a crucial role: the input layer receives the initial feature set, the hidden layers transform the features through a series of computations, and the output layer delivers the final classification result. Each hidden layer contains multiple neurons that activate or remain idle based on the inputs they receive, thereby influencing the forward propagation process. The architecture of the proposed DNN, depicted in Figure 2, consists of an input layer, four hidden layers, and a single output layer. The input layer takes the feature vector X derived from the feature selection process, while www represents the weight of the connection between neurons in Layer i to Layer $i+1$. The activation functions a determines whether a neuron should activate based on the calculations performed within each layer. Different activation functions are assigned to distinct layers to optimize the learning process: Rectified Linear Unit (ReLU) is applied within the hidden layers to efficiently handle non-linear transformations, while Softmax is used in the output layer to classify the detected attacks into distinct categories. The ReLU activation function in the hidden layers, as defined in Equation (15), ensures that neurons respond to positive values while discarding negative inputs, enhancing computational efficiency. Meanwhile, the Softmax activation function, shown in Equation (16), transforms the output layer's values into probabilities, allowing for a clear, interpretable output that indicates the likelihood of each possible attack class. This

robust DNN setup, with its carefully tuned parameters and layer-specific activation functions, provides an effective means of accurately detecting IoT-based attacks.

$$ReLU(a_i) = \max(0, a_i) \tag{15}$$

$$Softmax(a_i) = \frac{e^{a_i}}{\sum_{j=1}^{n} e^{a_i}}, \tag{16}$$

In a Deep Neural Network (DNN), each neuron performs calculations through two main stages: forward propagation and backward propagation. During forward propagation, each input value is multiplied by its respective weight, and a bias term is added to produce an intermediate output, which flows through the hidden layers. This process ultimately yields the predicted output, denoted as *Y*, at the final layer. In each hidden layer *L*, a neuron computes an output by applying an activation function to the weighted sum of its inputs, determining whether to activate and transmit its signal to the next layer. This process, layer by layer, refines the prediction, guiding it toward accuracy.

$$a_l = w_l^L . H^{l-1} + b^l \tag{17}$$

$$H_l = a(H_l), \tag{18}$$

In training the DNN, backpropagation is employed with gradient descent (GD) to adjust the weights iteratively, aiming to reduce the error between the model's predictions and the actual values. Starting at the output layer, the error is calculated and then backpropagated through each hidden layer, allowing for weight and bias adjustments based on the gradients. The gradients indicate the necessary direction and magnitude of changes to minimize error at each step. To enhance this process, the Adam optimizer is used, combining momentum-based gradient descent with Root Mean Square (RMS) propagation for faster and more stable convergence. Momentum helps by factoring in the average direction of previous gradients, enabling smoother and quicker progress towards minimizing error. RMS prop, meanwhile, applies a weighted average to the squared gradients (denoted as dw^2), which reduces oscillations by adjusting more conservatively in directions with larger gradients. The Adam optimizer leverages both momentum and RMS by calculating past squared gradients *U* and past momentum *V* using Equations (19) and (20). A bias-correction term is applied to both *U* and *V* via Equations (21) and (22), counteracting potential biases in the early stages of training. Finally, Equation (23) updates the weights, effectively combining these refined gradient adjustments. This approach provides a dynamically adaptive learning rate, improving the efficiency and accuracy of the DNN's training process across all layers.

$$U = \beta_1 V + (1 - \beta_1) dw \tag{19}$$

$$V = \beta_2 V + (1 - \beta_2) dw2 \tag{20}$$

$$U = \frac{U}{1 - \beta_1^i} \tag{21}$$

$$V = \frac{V}{1 - \beta_2^i} \tag{22}$$

$$w' = w - \alpha \frac{U}{\sqrt{V}+\epsilon}, \quad (23)$$

In this approach, α represents the learning rate, which controls the size of each step in gradient descent, while β, set between 0 and 1, adjusts the influence of past gradients for smoother convergence. The DNN model is then trained and evaluated on the BoT-IoT dataset, using optimized hyperparameters to boost classification accuracy and model performance.

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

The proposed attack detection model, which incorporates efficient feature selection, is implemented using Python's Scikit-Learn library. This section presents the experimental results evaluated through various metrics, including accuracy, precision, recall, and area under the curve (AUC). These metrics provide a comprehensive assessment of the model's effectiveness in accurately identifying attacks within IoT environments.

5.1 Evaluation Metrics

The performance of the proposed model for attack detection is evaluated using a confusion matrix, which categorizes the results into four main metrics: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). This matrix provides a comprehensive assessment of the model's effectiveness in accurately identifying both attack and non-attack scenarios in the IoT environment.

Table 3: Confusion Matrix

Actual class	**Predicted class**	
	Normal	**Attack**
Normal	True_negative (TN)	False_positive (FP)
Attack	False_negative (FN)	True_positive (TP)

The performance metrics used to evaluate the proposed attack detection model are crucial for understanding its effectiveness in identifying network traffic patterns.

True Positive (TP) refers to the number of instances that the network correctly identifies as belonging to the desired class (i.e., actual attacks). This metric highlights the model's ability to detect real threats.

True Negative (TN) indicates the number of instances that the network correctly classifies as not belonging to the desired class (i.e., benign traffic). A high TN count is essential for minimizing false alarms in an operational setting.

False Positive (FP) represents the instances that the model fails to recognize as belonging to the desired class. This metric is critical because it signifies missed detections, which can lead to security vulnerabilities.

False Negative (FN) captures the instances that the network accurately detects as not belonging to the desired class. Reducing FN is vital to ensure that genuine threats do not go unnoticed.

LogLoss, or logistic loss measures the accuracy of the method based on probabilistic outputs. A LogLoss value of 0 indicates perfect predictions, while higher values suggest increasing discrepancies between predicted probabilities and actual labels.

Training Time denotes the duration required to develop the classification model. This is an important factor to consider, as longer training times may affect the model's deployability and scalability in real-world applications. Balancing model complexity with training efficiency is key to achieving optimal performance.

5.2. Hyperparameter Configurations of the Suggested IoT Attack Detection Model

The proposed efficient feature selection integrated with a deep learning-based attack detection model employs ReLU and Softmax activation functions to enhance performance. The learning rate, a crucial hyperparameter, significantly influences model training. A high learning rate can lead to instability, preventing convergence, while a low learning rate may cause overfitting due to prolonged training without sufficient adjustment.

To identify the optimal learning rate, the model was assessed across a spectrum from 0 to 1. As shown in Figure 3, the model attained an impressive accuracy of 99.2% with a learning rate of 0.0015. This finding underscores that increasing the learning rate beyond this point resulted in decreased accuracy. Therefore, 0.0015 was established as the optimal learning rate for the model.

Regarding the number of epochs, the training was conducted with a maximum limit of 100 epochs. However, it was noted that the training loss stabilized after 60 epochs, indicating that additional training would not yield significant improvements. As a result, the epoch count was set at 60 for the proposed model, optimizing training efficiency while minimizing the risk of overfitting.

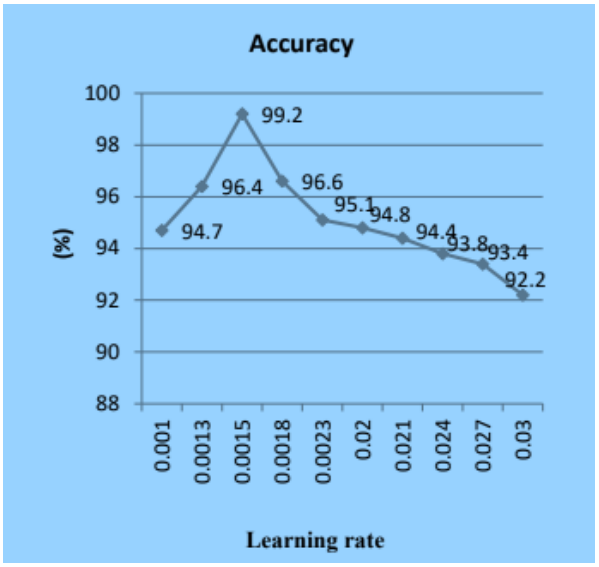

Fig. 2. Various learning rates affect accuracy.

5.3. Result Analysis

The confusion matrix for the proposed attack detection model is illustrated in Table 4, highlighting its performance metrics. The evaluation results using the BoT-IoT dataset are detailed in Table 5. These findings underscore the model's effectiveness, proving its ability to accurately detect attacks in the IoT environment, thereby validating its efficiency and reliability.

Table 4: Confusion matrix evaluation

Actual class	Predicted class	
	Normal	Attack
Normal	99.7	0.3
Attack	1.5	98.4

Figure 4 showcases the detection rates for different attack classes in the dataset, illustrating the efficacy of the proposed model. It achieved notable detection rates of 98.6% for Reconnaissance, 98.7% for DoS, 96.7% for DDoS, 98.8% for Information Theft, and 99.1% for Normal instances. These outcomes indicate that the model is highly effective in accurately detecting both attack and normal classes, highlighting its robustness in an IoT environment.

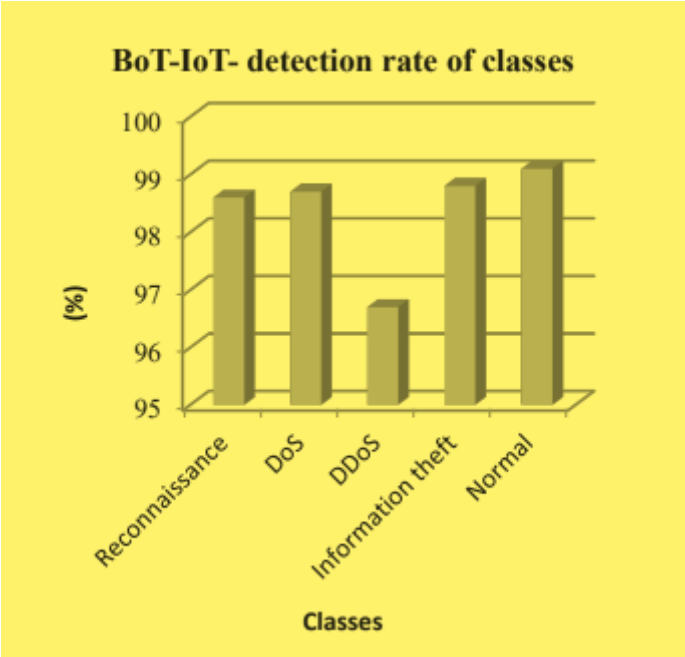

Fig.3. Rate of BoT-IoT dataset class detection

The performance of the proposed DTPCRFE-ODNN model has been rigorously assessed against various existing feature selection models, including Wrapper-based Neuro Tree, Knowledge Graph, Improved Principal Component Analysis, and Modified kNN. As shown in Table 7, the proposed model achieved an outstanding accuracy of 99.2%, surpassing the performance of traditional feature selection models. The results were evaluated during both training and testing phases, revealing only minor variations, which highlights the model's stability and reliability. In addition to high accuracy, the DTPCRFE-ODNN model demonstrated improved metrics, including a lower false prediction rate and enhanced F-score, recall, and precision values.

Table 5: Comparison of suggested and current algorithms

Feature selection model	BoT-IoT					
	Phase	Accuracy (%)	FPR (%)	F-Score (%)	Recall (%)	Precision (%)

Wrapper - Neurotree	Training	93.37	1.86	95.67	94.57	92.86
	Testing	93.16	1.76	95.14	94.38	92.74
Knowledge Graph	Training	96.48	1.67	96.77	96.47	94.88
	Testing	96.74	1.63	96.66	96.54	94.57
Improved PCA	Training	96.49	1.98	97.176	96.77	95.98
	Testing	96.83	1.93	97.23	97.65	95.87
Modified kNN	Training	94.25	1.57	96.27	96.37	94.96
	Testing	94.34	1.56	96.33	96.03	94.24
Proposed DT-PCRFE	Training	99.07	1.05	98.88	98.96	98.67
	Testing	99.13	1.06	98.95	98.67	98.44

Moreover, the effectiveness of the proposed feature selection combined with classification was evaluated against other classifiers such as CNN-BiLSTM, CNN-DBN, and Auto Encoder with DNN. This comparative analysis focused on detection rates and Area Under the Curve (AUC), with results illustrated in Figures 5 and 6. The DTPCRFE-ODNN model secured a remarkable detection rate of 99.1% and an AUC of 0.99, indicating optimal performance compared to alternative detection systems. In comparison, existing approaches performed at lower levels: CNN-BiLSTM achieved a detection rate of 98.3% with an AUC of 0.976; CNN-DBN recorded a detection rate of 98.6% and an AUC of 0.98; while the Auto Encoder with DNN attained a detection rate of 97.3% and an AUC of 0.97. The superior performance of the DTPCRFE-ODNN model is attributed to its effective feature selection and fusion techniques, along with an optimized deep neural network architecture. This positions the proposed attack detection system as a robust solution for identifying attacks in IoT environments, showcasing its potential to significantly enhance security measures in this rapidly evolving field.

5. CONCLUSION

The widespread adoption of IoT-enabled systems has exposed significant security vulnerabilities, primarily due to the advanced technologies that facilitate these environments. While the inherent characteristics of IoT—such as efficiency, functionality, and versatility—offer numerous benefits, they also make these systems susceptible to malicious exploitation of sensitive information. To mitigate these security threats, this paper introduces an innovative feature selection and detection system designed specifically for IoT environments, utilizing both machine learning (ML) and deep learning (DL) methodologies.

REFERENCES

[1]. Krishna, E. S., Thangavelu, A. Attack Detection in IoT Devices Using Hybrid Metaheuristic Lion Optimization Algorithm and Firefly Optimization Algorithm. International Journal of System Assurance Engineering and Management, 2021, 1-14. https://doi.org/10.1007/ s13198-021-01150-7.

[2]. Muna, A. H., Moustafa, N., Sitnikova, E. Identification of Malicious Activities in Industrial Internet of Things Based on Deep Learning Models. Journal of Information Security and Applications, 2018, 41, 1-11. https://doi. org/10.1016/j.jisa.2018.05.002.

[3]. Yang, X., Peng, G., Zhang, D., Lv, Y. An Enhanced Intrusion Detection System for IoT Networks Based on Deep Learning and Knowledge Graph. Hindawi Security and Communication Networks, 2022, Article ID 4748528. https://doi.org/10.1155/2022/4748528.

[4]. Zhou, Y., Qin, R., Xu, H., Sadiq, S., Yu, Y. A Data Quality Control Method for Seafloor Observatories: The Application of Observed Time Series Data in the East China Sea. Sensors, 2018, 18, 2628. https://doi.org/10.3390/ s18082628.

[5]. Akbulut, Y., Şendağ, S., Birinci, G., Kılıçer, K., Şahin, M. C., & Odabaşı, H. F. (2008). Exploring the Types and Reasons of Internet-triggered Academic Dishonesty Among Turkish Undergraduate Students: Development of Internet-Triggered Academic Dishonesty Scale (ITADS). *Computers & Education*, 51(1), 463–473. https://doi.org/10.1016/j.compedu.2007.06.003.

[6]. Li, X. (2006). Using peer review to assess coding standards–a case study. In Frontiers in education conference, 36th annual (pp. T2D-9–T2D-14). San Diego, CA, USA.

[7]. Kumar, R., Singh, J.P., Srivastava, G. (2014). Altered Fingerprint Identification and Classification Using SP Detection and Fuzzy Classification. In: , et al. Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012), December 28-30, 2012. Advances in Intelligent Systems and Computing, vol 236. Springer, New Delhi. https://doi.org/10.1007/978-81-322-1602-5_139

[8]. Li, X. (2007). Incorporating a code review process into the assessment. In 20th annual conference of the National Advisory Committee on Computing Qualifications (NACCQ 2007) (pp. 125–131). Nelson, New Zealand

[9]. Pirretti, M., Zhu, S., Vijaykrishnan, N., McDaniel, P., Kandemir, M., Brooks, R. The Sleep Deprivation Attack in Sensor Networks: Analysis and Methods of Defense. International Journal of Distributed Sensor Networks, 2006, 2(3), 267-287. https://doi.org/10.1080/15501320600642718.

[10]. Brun, O., Yin, Y., Gelenbe, E., Kadioglu, Y. M., Augusto-Gonzalez, J., Ramos, M. Deep Learning with Dense Random Neural Networks for Detecting Attacks Against IoT-connected Home Environments. In International ISCIS Security Workshop, Springer, Cham, 2018, 79-89. https://doi.org/10.1007/978-3-319-95189-8_8

[11]. Kan, X., Fan, Y., Fang, Z., Cao, L., Xiong, N. N., Yang, D., Li, X. A Novel IoT Network Intrusion Detection Approach Based on Adaptive Particle Swarm Optimization Convolutional Neural Network. Information Sciences, 2021, 568, 147-162. https://doi.org/10.1016/j. ins.2021.03.060.

[12]. Alexandron, G., Yoo, L. Y., Ruipérez-Valiente, J. A., Lee, S., & Pritchard, D. E. (2019). Are MOOC Learning Analytics Results Trustworthy? With Fake Learners, They Might Not Be! *International Journal of Artificial Intelligence in Education*, 29(4), 484–506. https://doi.org/10.1007/s40593-019-00183-1

[13]. Bar,Y, et al.(2015): Chest pathology detection using deep learning with non-medical training. In 2015 IEEE 12th international symposium on biomedical imaging (ISBI), pages 294–297, IEEE