IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Intrusion Detection System (IDS) using Machine Learning Decision Tree Classification Algorithm on NSL-KDD Cup Dataset

Aditya Kumar Singh[1]; Ratul Tapader[1]; Debanjali Biswas[1]; Sudipta Roy[1]; Sabyasachi Samanta[1]

[1]Haldia Institute of Technology, Department of CSE (Cyber Security) Haldia, Purba Medinipur, West Bengal, India 2024, Pin- 721657

Abstract: Effective intrusion detection systems (IDS) rely on comprehensive and realistic data sets that emulate real-world network events. Traditionally, the KDD-CUP 99 data set has been utilized for this purpose; however, it has been criticized for its limitations, leading to the adoption of the improved NSL-KDD data set. This research investigates the performance of various classification algorithms and data mining techniques, including decision trees, in identifying anomalies within network traffic using the NSL-KDD data set. We explore the interplay between network protocols in the protocol stack and the intrusion tactics employed by malicious actors to generate unusual network patterns. The analysis leverages the capabilities of the data mining tool WEKA 3.9.5 for preprocessing and preliminary examination, while Python is employed for implementing the classification algorithms. Experimental results indicate that our proposed model, particularly using decision trees, demonstrates high efficacy and robustness, achieving an accuracy of 0.997% as evaluated through metrics such as precision, false-positive rate. Our study begins by detailing the shortcomings of the KDD-CUP 99 data set and the enhancements introduced with the NSL-KDD data set. We then describe the preprocessing steps undertaken using WEKA 3.9.5 to prepare the data for analysis. Various classification algorithms, including decision trees, support vector machines, and neural networks, are implemented in Python to classify network traffic. The performance of these algorithms is rigorously evaluated, with a particular focus on their ability to accurately distinguish between normal and anomalous traffic. By understanding these relationships, we aim to enhance the detection capabilities of IDS. The results of our experiments are promising. The proposed model not only achieves high accuracy and precision but also maintains a low false-positive rate, which is crucial for practical deployment in real-world networks. The integration of WEKA for data preprocessing and Python for implementing sophisticated classification algorithms showcases a powerful approach to detecting network anomalies. Future research will focus on refining these models and exploring additional data sets to further enhance the detection performance of IDS, ensuring they remain effective against evolving cyber threats. By continuously updating the data sets and refining the models, we can better anticipate and counteract the innovative strategies employed by cyber attackers. This ongoing improvement will help maintain the integrity and security of network systems in an increasingly interconnected and vulnerable digital landscape.

Keyword: Intrusion Detection System; Decision Tree; Machine Learning; WEKA; Statistical Analysis. Support Vector Machine**;** Principal Component Analysis**;** Confusion matrix**;** Random Forest; NSL – KDD.

Introduction: In our modern age, communication systems form the backbone of everyday life, facilitating a myriad of essential functions across diverse sectors. Computer networks have become instrumental in processing corporate data, enabling educational endeavours, fostering collaborative teamwork, acquiring vast data repositories, and offering entertainment avenues. However, the pervasive utilization of these networks

has introduced a complex web of vulnerabilities within the existing communication protocol stack [1]. Computer networks are essential to many facets of daily life in the current era. They constitute the foundation for data collecting, educational initiatives, cooperative teamwork, business data processing, and entertainment. These networks offer smooth teamwork through project management tools and real-time sharing of files, assist online learning and global study collaboration, and automate and streamline business procedures. In addition, they support the gathering and examination of enormous amounts of data, generating insights via big data and Internet of Things (IoT) applications and giving users access to digital entertainment via streaming services and online gaming portals [2]. However, a complicated web of vulnerabilities inside the communication protocol stack is introduced by the increased reliance on these networks. Phishing schemes and virus dissemination pose a threat to users at the application layer because they take advantage of system vulnerabilities and user trust to steal confidential data or interfere with operations. The integrity and secrecy of data transmissions can be jeopardized by man-in-the-middle attacks and TCP/IP exploits, which can affect the transport layer. The network layer is vulnerable to Distributed Denial of Service (DDoS) attacks, which overwhelm network resources and cause disruptions and outages, as well as IP spoofing, in which attackers pretend to be someone else [3].

ARP poisoning and MAC spoofing can be used to compromise the data link layer further down, giving attackers the ability to circumvent network security measures and intercept or reroute data. The physical layer is also vulnerable to threats like signal jamming, which interferes with wireless communications, and wiretapping, which results in data breaches through physical interception of network cables. Implementing strong mitigation measures is crucial to fighting these security issues. Improved encryption methods safeguard secrecy and data integrity at every connection layer. Software and firmware patches and updates on a regular basis fix known vulnerabilities and lower the chance of exploitation. While network segmentation isolates important portions, limiting the potential spread of breaches, intrusion detection and prevention systems (IDPS) assist in identifying and thwarting harmful actions. Finally, a key component of preserving network security is teaching users about safe habits and how to spot possible dangers. Through comprehension and resolution of these susceptibilities, we may guarantee the sustained advantages of networked communication systems while mitigating possible risks, consequently upholding the authenticity and safety of our globalized society [4]. The foundational intent behind the construction of the current protocol stack was to ensure transparency and user-friendliness. Yet, this very adaptability has inadvertently exposed it to potential assaults orchestrated by malicious actors. Consequently, the need for continuous surveillance and fortified security measures to protect computer networks has become paramount. At the forefront of network defence mechanisms stands the Intrusion Detection System (IDS), a critical component automating the surveillance and monitoring of network activities. An IDS, comprising both hardware and software elements, undertakes the task of meticulously scanning networks for any suspicious or malicious activities and promptly alerting system administrators. It acts as a vigilant sentinel, complementing traditional security protocols such as access restrictions and authentication mechanisms [5].

The Intrusion Detection System (IDS), a vital component that automates the surveillance and monitoring of network activities, is at the forefront of network security systems. An intrusion detection system (IDS), which consists of hardware and software components, is responsible for carefully monitoring networks for any unusual or harmful activity. If something is found, it notifies system administrators right once. Serving as a watchful guardian, it supplements established security measures like access controls and authentication procedures, greatly augmenting their effectiveness. An essential layer of defence is provided by an intrusion detection system (IDS), which continuously monitors network traffic and analyses it for indications of potential attacks. It can spot odd trends, illegal access attempts, and other kinds of cyberattacks like phishing, malware, and denial-of-service (DoS) attacks. By using a proactive approach to network security, communication systems' availability, integrity, and confidentiality are protected from major intrusions through early risk identification and mitigation [6].

IDS plays an increasingly important role in protecting digital infrastructure as cyber threats continue to increase in complexity and frequency. It offers comfort and makes it possible for businesses to run safely in a globalized society. Organizations may create a strong defence-in-depth approach that drastically lowers the risk of cyber events and improves their overall cybersecurity posture by combining an IDS with additional security measures. Adding an intrusion detection system (IDS) to a network architecture has many advantages

over only detecting threats. It offers insightful information on user behaviour and network activities that can be utilized to improve general security procedures and policies. By guaranteeing that security monitoring is in place and that occurrences are recorded and reported, it also helps with regulatory compliance. The fundamental significance of an IDS lies in its role as a second line of defence, dedicated to preserving the confidentiality, accessibility, and integrity of the most prized assets: information. Nevertheless, the effectiveness of an IDS is contingent upon a comprehensive understanding of the prevailing infrastructure and the authentic requisites of the system owners before investing resources into such a security measure [7].

In this comprehensive study, we turn our attention to the NSL-KDD dataset—a cornerstone of modern research in intrusion detection systems. Leveraging a diverse array of classification methods and attribute selection techniques available in Python 3.10's classification tool and the WEKA 3.9.5 data mining tool, we endeavour to delve deep into the analysis of this dataset. Our methodology involves segmenting the NSL-KDD dataset into four distinct clusters, each representing prevalent attack types: Denial of Service (DoS), Probe, User-to-Root (U2R), and Remote-to-Local (R2L). Through meticulous analysis of both test and training datasets, we aim to evaluate the accuracy and computational efficiency of various classification algorithms in identifying and categorizing these attack types [8]. The central objective of this project is to unravel the intricate ways in which intrusion detection systems function in detecting different types of assaults, while simultaneously gauging the efficacy of these systems in fortifying and safeguarding networks. By offering an in-depth analysis, we aspire to furnish insights that can bolster the performance of intrusion detection systems, ensuring robust and reliable security measures without compromising the functionality of networks [9].

Related Work: TNN-IDS: Transformer neural network-based intrusion detection system for MQTT-enabled IoT Networks [1] it says that the Internet of Things (IoT) connects smart devices globally, but its openness makes it vulnerable to security threats. MQTT, a popular protocol in IoT, faces risks like eavesdropping and attacks. To counter this, a Transformer Neural Network-based Intrusion Detection System (TNN-IDS) was developed. TNN-IDS harnesses Transformer NN's parallel processing, boosting the detection of malicious activities in MQTT-enabled IoT networks. Compared to other systems, TNN-IDS showcases superior performance, achieving a remarkable 99.9% accuracy in spotting threats. This innovation marks a significant step in fortifying IoT security, potentially ensuring safer data handling in these interconnected networks.

Metaverse-IDS: Deep learning-based intrusion detection system for Metaverse-IoT networks The fusion of the metaverse and the Internet of Things (IoT) heralds a new era of interconnected, virtual networks. This convergence promises to create meaningful links between physical and virtual spaces, allowing real-time data processing and analysis.

Yet, the amalgamation of these networks brings forth a multitude of security and privacy risks. Addressing these concerns, a groundbreaking intrusion detection system (IDS) model has emerged, utilizing cutting-edge deep learning techniques. This IDS model employs Kernel Principal Component Analysis (KPCA) for extracting attack features and Convolutional Neural Networks (CNN) for recognizing and classifying attacks within the metaverse-IoT communication landscape.

Extensive evaluation using industry-standard datasets like BoT-IoT and ToN-IoT, encompassing various IoT attacks relevant to metaverse-IoT interactions, showcased the IDS's prowess. It successfully identified 12 distinct attack classes, boasting an impressive accuracy rate while maintaining a low False Negative Rate (FNR).

Comparative analysis against existing models revealed the superior performance of this IDS, highlighting its potential as a robust solution for fortifying the security posture of interconnected metaverse-IoT networks. This innovative approach marks a significant leap forward in safeguarding the integrity and safety of these complex, interwoven realms.

An optimized Intrusion Detection Framework using Deep Transfer Learning and Genetic Algorithm [3] In the realm of the Industrial Internet of Things (IIoT), the escalating vulnerability to cyber threats demands advanced Intrusion Detection Systems (IDSs). Our research contributes to cybersecurity by introducing an optimized IDS based on Deep Transfer Learning (DTL), tailored for diverse IIoT networks. Our framework leverages Convolutional Neural Networks (CNNs), Genetic Algorithms (GA), and ensemble techniques in a tri-layer architecture. First, we convert the Edge_IIoTset dataset into image data for CNN-based analysis. Next, GA fine-tunes model hyperparameters, enhancing adaptability. Finally, ensemble techniques merge top-performing models, fortifying IDS robustness. Rigorous evaluation validated our framework, achieving 100% attack detection accuracy against 14 cyberattack types. This showcases its efficacy in bolstering security for IIoT networks, offering crucial insights for future adaptive IDS development.
New automatic (IDS) in IoTs with artificial intelligence technique [4] the use of Wireless Sensor Networks (WSNs) has expanded across surveillance, monitoring, and home automation, with applications in industries. However, WSNs face unique challenges, notably security issues. Intrusion Detection Systems (IDS) and Security Information and Event Management (SIEM) systems play key roles in monitoring and alerting for unauthorized access or breaches.
Securing these networks from attacks exploiting IoT devices has led to the exploration of Artificial Intelligence (AI) for packet analysis and pattern recognition. However, resource limitations on IoT devices hinder the implementation of complex AI methods like Artificial Neural Networks (ANNs) for Intrusion Detection due to their computational demands.

Yet, innovative approaches like Neuro-Evolution for Augmenting Topology (NEAT) using Genetic Algorithms show promise in optimizing neural networks for Reinforcement Learning environments with reduced complexity but high performance. The study presented four automatic systems addressing these challenges and proposed solutions. It involved an introduction, surveying related techniques, developing a prototype, and concluding with validation using a common dataset to assess results. Intrusion detection system using transformer-based transfer learning for imbalanced network traffic [5] The IDS-INT system revolutionizes network intrusion detection by merging transformer-based transfer learning, SMOTE for data balance, and a CNN-LSTM hybrid model. It meticulously captures attack details, employs cutting-edge learning methods, and extracts deep features to spot and categorize attacks. Through rigorous testing on established datasets and a commitment to explainable AI, IDS-INT offers a powerful, transparent solution for safeguarding against network threats.

METHODOLOGY:

Random Forest Classifier: It is a powerful machine learning technique used for both classification and regression tasks.

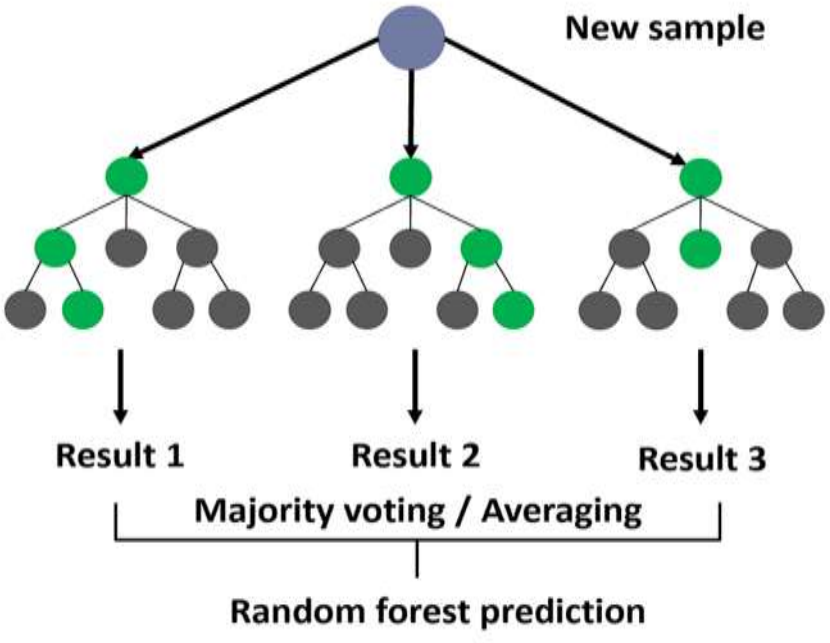

Figure 1: generative structure random forest classifier.

- Bagging (Bootstrap Aggregating)

Random Forests use bagging to create different subsets of the training data.

- Bootstrap Sample: For each tree, a random sample (with replacement) of the training set is taken. If the training set has n samples, the bootstrap sample will also have n samples, but some will be repeated, and some original samples will be missing.
- Random Feature Selection

To add randomness, each tree is allowed to use only a random subset of features for splitting at each node. This ensures the trees are diverse.

- If there are mmm features in total, a typical choice is to use $\sqrt{m}$ features for classification tasks and m/3 features for regression tasks.
- Constructing the Forest

Training: Repeat the following for B trees:

- Create a bootstrap sample from the training data.
- Grow a decision tree from the bootstrap sample. At each node, select the best split from a random subset of features.
- Continue splitting until a stopping criterion (e.g., maximum depth or minimum samples per leaf) is met.

Prediction:

- For classification: Each tree gives a class prediction, and the forest predicts the class with the most votes (majority voting).
- For regression: Each tree gives a numerical prediction, and the forest predicts the average of these predictions.

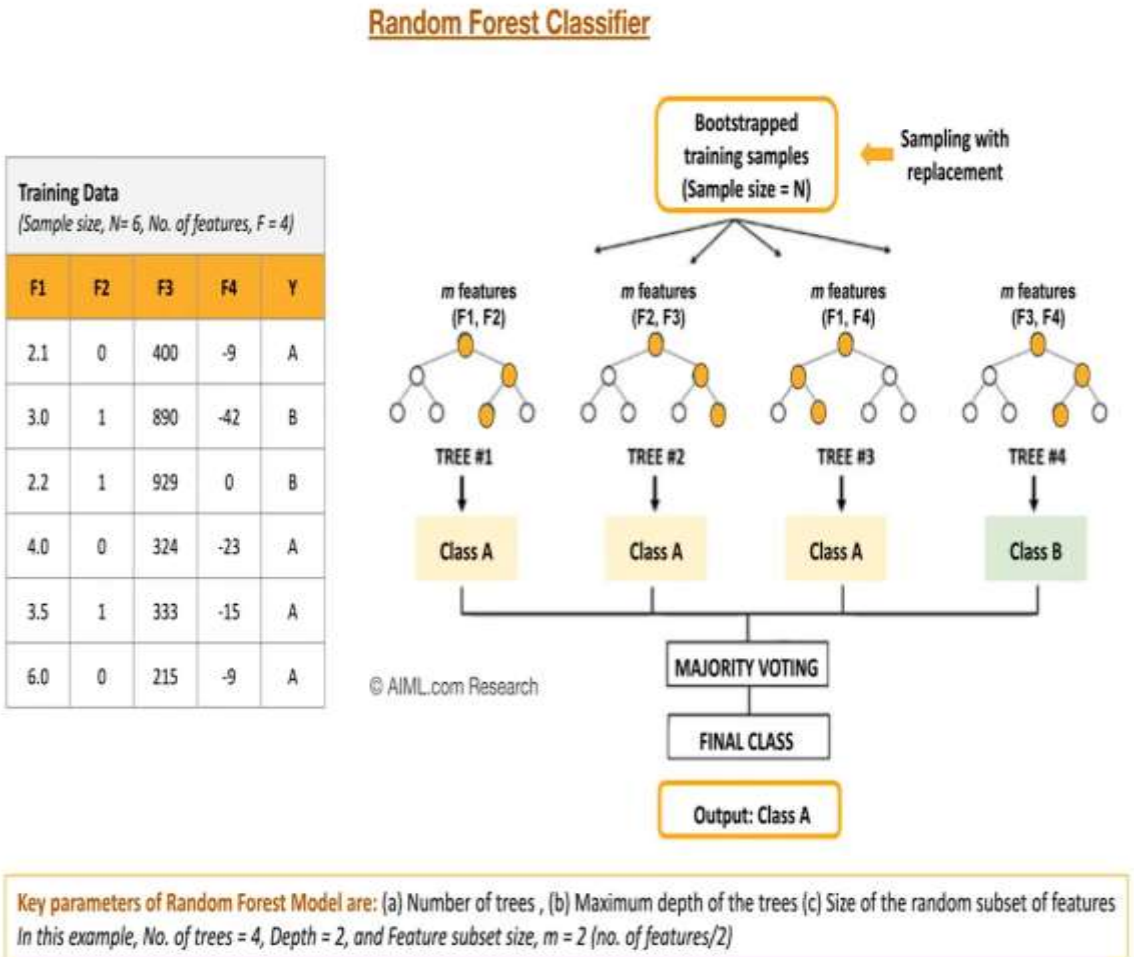

Figure 2: Random Forest Classifier

Random forests are a powerful ensemble learning method used for classification, regression, and other predictive tasks. They function by constructing a multitude of decision trees during the training phase and outputting either the mode of the classes (in classification) or the average prediction (in regression) from these individual trees. The process begins with bootstrap sampling, where multiple subsets of the training data are created through random sampling with replacement. For each of these subsets, a decision tree is constructed, but rather than considering all features at each node, only a random subset of features is selected, and the best split is chosen from this subset. This introduction of randomness helps in reducing overfitting and ensures that the trees in the forest are diverse[10].

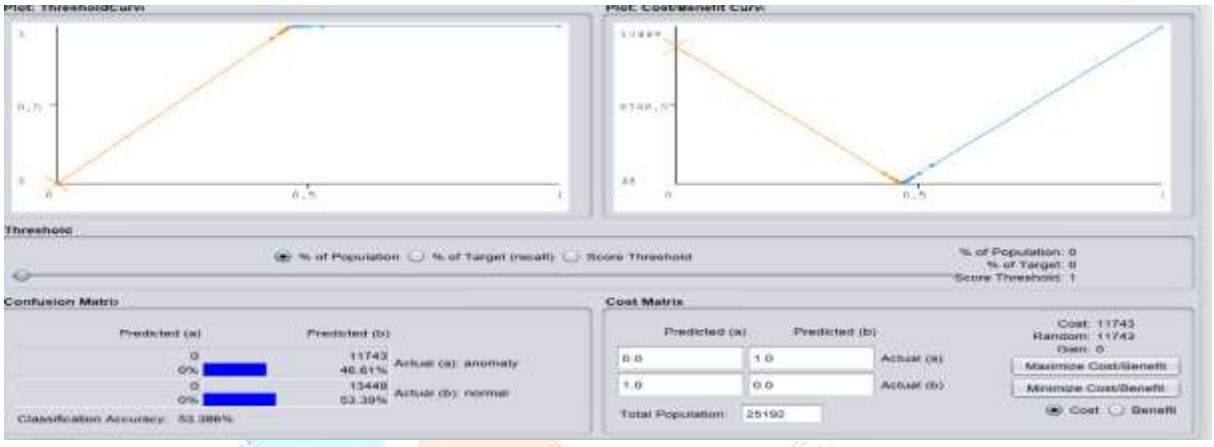

Figure 3: Threshold Curve and Cost/Benefit Curve

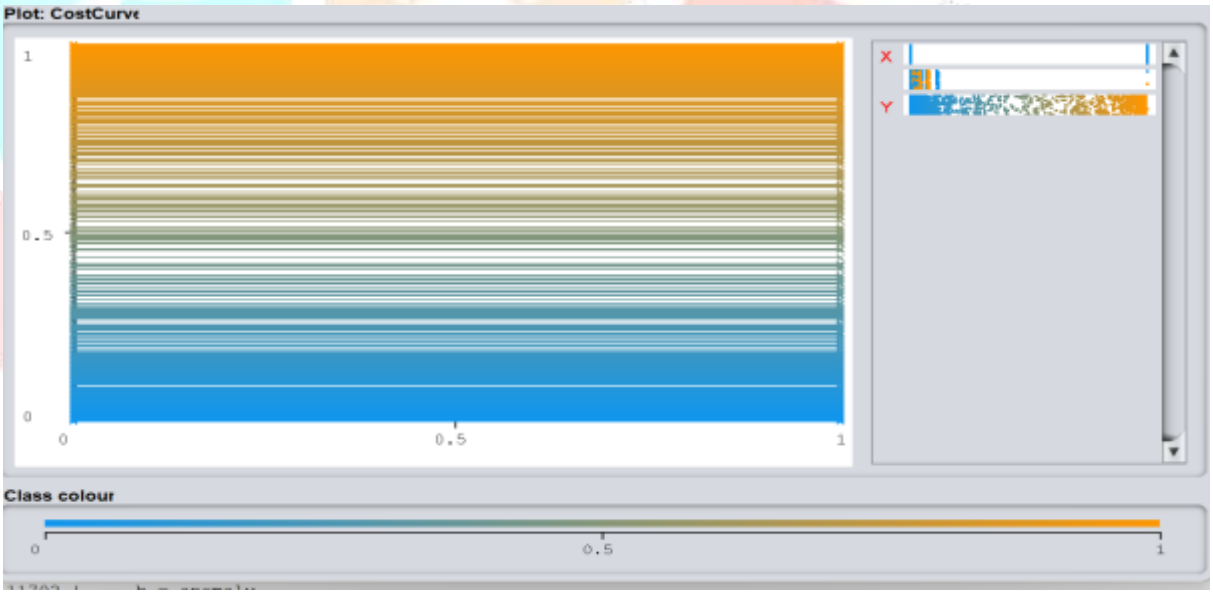

Figure 4: Cost Curve of Random Forest classifier

One of the key advantages of random forests is their ability to aggregate the predictions of multiple trees, thus reducing the risk of overfitting that individual decision trees often face. The random selection of features at each split, known as the random subspace method, contributes to this by ensuring the trees are less correlated. Additionally, the out-of-bag (OOB) error method provides an unbiased estimate of the prediction error. Since each tree is trained on different data subsets, the data not included in the bootstrap sample can be used for validation, serving as a built-in cross-validation mechanism. Random forests are widely applied in various fields for tasks such as classification and regression due to their robustness and effectiveness[18]. By leveraging the strengths of multiple decision trees, they offer improved accuracy and generalization compared to single-tree models, making them a go-to choice for many predictive modelling challenges. They are great for handling large datasets, dealing with missing values, and avoiding overfitting that might occur with a single decision tree. They have found applications in various fields such as finance, healthcare, and image recognition due to their accuracy and versatility. Random forests, a versatile and robust ensemble learning method, find extensive applications across diverse real-world domains. Healthcare leverages their predictive capabilities for disease diagnosis and prognosis, where they analyse patient data to predict the likelihood of

various illnesses such as diabetes, cancer, or cardiovascular conditions. Moreover, in medical imaging, random forests aid in the classification of images to detect anomalies like tumours, aiding in early diagnosis and treatment planning.

Decision Trees: A decision tree is the fundamental building block of a random forest. It splits the data based on feature values to create branches, leading to decisions or classifications. Decision Trees are hierarchical structures that recursively partition the data space into smaller regions based on feature values. They are popular due to their simplicity and interpretability, as they mimic human decision-making processes[19]. A decision tree consists of nodes, branches, and leaves:

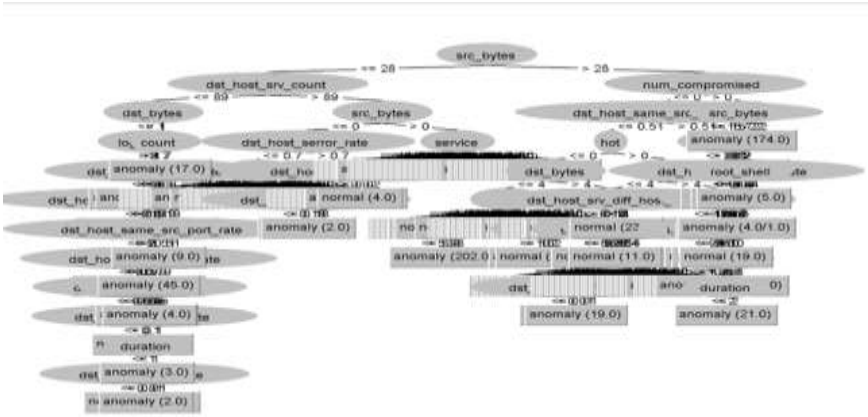

Figure 5: Tree view of our proposed Decision Tree.

- **Nodes**: Represent decision points based on feature values.
- **Branches**: Connect nodes and represent the decision outcomes.
- **Leaves (or terminal nodes)**: Represent the final decision or outcome.

Impurity	Task	Formula	Description
Gini impurity	Classification	$\sum_{i=1}^{C} f_i(1-f_i)$	f_i is the frequency of label i at a node and C is the number of unique labels.
Entropy	Classification	$\sum_{i=1}^{C} -f_i \log(f_i)$	f_i is the frequency of label i at a node and C is the number of unique labels.
Variance / Mean Square Error (MSE)	Regression	$\frac{1}{N}\sum_{i=1}^{N}(y_i-\mu)^2$	y_i is label for an instance, N is the number of instances and μ is the mean given by $\frac{1}{N}\sum_{i=1}^{N} y_i$
Variance / Mean Absolute Error (MAE) (Scikit-learn only)	Regression	$\frac{1}{N}\sum_{i=1}^{N}\lvert y_i-\mu\rvert$	y_i is label for an instance, N is the number of instances and μ is the mean given by $\frac{1}{N}\sum_{i=1}^{N} y_i$

Figure 6: Mathematical Foundation of Decision Tree.

Decision Tree Construction

1. Splitting Criteria

At each node of a decision tree, a splitting criterion is used to decide how to split the data into subgroups. The commonly used criteria include:

- Gini Impurity: Measures the impurity or disorder in a node for classification tasks.
- Entropy: Measures the information gain in a node for classification tasks.
- Variance Reduction: Measures the reduction in variance for regression tasks.

2. Recursive Partitioning

The tree grows recursively by splitting nodes until a stopping criterion is met, such as:

- Maximum depth of the tree.
- Minimum number of samples required to split a node.
- Minimum number of samples required to be at a leaf node.

3. Handling Categorical and Numerical Data
Decision trees can handle both categorical and numerical data:

- Categorical Data: Splitting is based on equality or inequality.
- Numerical Data: Splitting is based on thresholds.

4. Tree Pruning
To prevent overfitting, decision trees can be pruned after construction:

- Pre-pruning: Stop growing the tree early (e.g., based on depth or minimum samples per leaf).
- Post-pruning (or Pruning): Remove nodes or branches that do not provide additional predictive power.

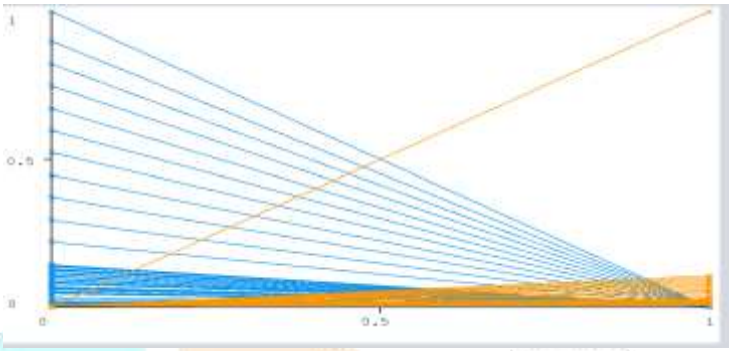

Figure 7: Visualized Cost Curve Anomaly

Advantages of Decision Trees

1. Interpretability: Easy to understand and interpret, suitable for non-experts.
2. Handling Non-linear Relationships: Can capture complex relationships between features and the target variable.
3. Handling Mixed Data Types: Can handle both numerical and categorical data without requiring preprocessing.
4. Feature Selection: Automatically selects important features for decision-making.
5. Scalability: Efficient for small to medium-sized datasets and quick to train.

Disadvantages of Decision Trees

1. Overfitting: Prone to overfitting, especially with deep trees and noisy data.
2. Instability: Small variations in the data can lead to different tree structures.
3. Bias Towards Dominant Classes: In classification, can create biased trees if one class dominates.
4. Difficulty in Capturing Linear Relationships: May not perform well with linearly separable data.

Evaluating Decision Trees

1. Metrics for Classification

- Accuracy: Proportion of correct predictions.
- Precision and Recall: Trade-offs between false positives and false negatives.
- F1-score: Harmonic mean of precision and recall.
- Confusion Matrix: Summary of true positives, true negatives, false positives, and false negatives.

2. Metrics for Regression

- Mean Absolute Error (MAE): Average of absolute differences between predicted and actual values.
- Mean Squared Error (MSE): Average of squared differences between predicted and actual values.
- R-squared (Coefficient of Determination): Proportion of the variance in the dependent variable that is predictable.

Support Vector Machine: Support Vector Machines (SVMs) are powerful supervised learning models used for both classification and regression tasks. They aim to find the optimal hyperplane that best separates different classes in the input space[20]. Some important factor of SVM is given bellow:

- Optimal Hyperplane: SVMs aim to find the hyperplane that best separates different classes in the input space. In a binary classification scenario, this hyperplane maximizes the margin, which is the distance between the hyperplane and the nearest data points from each class (support vectors). Maximizing this margin aids in better generalization and reduces the risk of overfitting.
- Support Vectors: These are the data points closest to the hyperplane and significantly influence its position and orientation. They are crucial in defining the decision boundary and determining the optimal hyperplane. SVMs primarily rely on these support vectors to make classifications.
- Kernel Trick: SVMs can handle non-linearly separable data by using kernel functions. Popular kernel functions like polynomial, radial basis function (RBF), or sigmoid enable SVMs to handle complex decision boundaries in the original feature space.

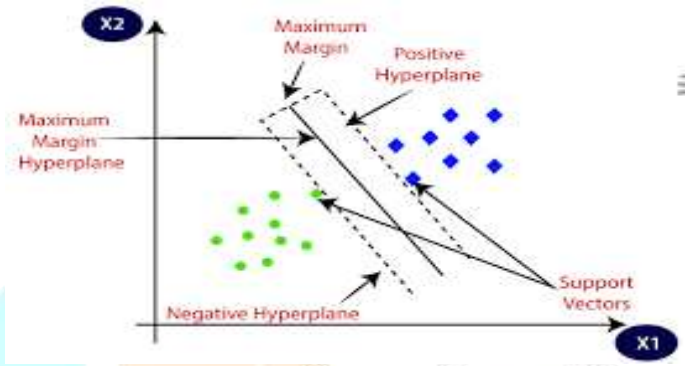

Figure 8: Pictorial representation of Proposed SVM.

The training of SVMs involves finding the optimal hyperplane by solving an optimization problem. This problem aims to maximize the margin while minimizing classification errors (soft margin SVM) or allowing for some misclassifications based on a chosen penalty (C parameter in the soft margin SVM). Techniques like quadratic programming are often used to solve this optimization problem efficiently[11].

Strengths and Applications:

- High-Dimensional Spaces: SVMs excel in scenarios where the number of dimensions is greater than the number of samples, making them suitable for text classification, image recognition, and genomic data analysis where feature spaces can be high-dimensional.

- Robustness: Due to the emphasis on maximizing the margin, SVMs tend to be robust against outliers in the training data.

- Versatility: They can handle both linear and non-linear decision boundaries using appropriate kernel functions.

- Resistance to Outliers: Random forests are robust to outliers and noisy data due to the averaging of predictions from multiple trees. Outliers have minimal impact on the overall model performance, making random forests suitable for datasets with varying degrees of data quality.

- Handle large datasets: Random forests can effectively handle large datasets with high dimensionality. They are parallelizable and can be trained efficiently on multi-core processors, making them suitable for big data applications.

Challenges:

- Computational Complexity: Training SVMs can become computationally expensive, especially with large datasets, as the time complexity scales with the square of the dataset size.

- Choosing the Right Kernel: Selecting the appropriate kernel and its parameters is crucial, and this choice can significantly impact the model's performance.

- Scalability: SVMs may not scale well to large datasets due to their computational complexity. Techniques like stochastic gradient descent (SGD) or using linear SVMs with approximate kernel methods are often employed to handle large-scale data.

- Kernel Section: Choosing an appropriate kernel function is crucial for achieving good performance with SVMs. However, selecting the right kernel can be challenging, as different kernels may perform differently depending on the dataset and problem at hand[12].

SVMs have been widely applied in various domains, including but not limited to text and sentiment analysis, bioinformatics, image recognition, and financial forecasting due to their flexibility, robustness, and ability to handle diverse data types and complexities. Support Vector Machines (SVMs) are a powerful class of supervised learning algorithms used for classification, regression, and outlier detection tasks. They are particularly well-suited for scenarios where the data is high-dimensional and the number of features exceeds the number of samples[13].

Confusion matrix: A confusion matrix is an essential tool in the realm of machine learning and statistical classification. It provides a comprehensive understanding of how well a classification model performs, especially in distinguishing between different classes. The matrix contrasts the actual target values with the model's predictions, enabling a clear view of how many instances were classified correctly and incorrectly.

The confusion matrix is essentially a table that allows us to evaluate the performance of a classification model. It looks like this:

1. True Positive (TP): The number of instances correctly predicted as positive.

2. False Positive (FP): The number of instances incorrectly predicted as positive.

3. True Negative (TN): The number of instances correctly predicted as negative.

4. False Negative (FN): The number of instances incorrectly predicted as negative.

For instance, in a medical diagnostic test where the task is to identify whether a patient has a disease (positive) or not (negative), the confusion matrix helps in understanding how many true cases were correctly identified, how many were missed, and how many false alarms were raised. The confusion matrix is significant for several reasons[14]:

Comprehensive Evaluation: It provides a complete picture of how the model is performing by showing not just the accuracy, but also where the model is making errors.

1. Error Analysis: By examining the confusion matrix, one can identify specific areas where the model struggles, such as a particular class that is frequently misclassified[15].

2. Metric Derivation: Various performance metrics like accuracy, precision, recall, specificity, and the F1 score are derived from the confusion matrix, offering a nuanced evaluation of the model.

3. Improvement Insights: Insights from the confusion matrix can guide improvements in the model, such as adjusting thresholds, tuning parameters, or collecting more data for specific classes.

Metrics Derived from the Confusion Matrix

Several important metrics can be calculated from the confusion matrix, each offering unique insights into the performance of the classification model[16].

Accuracy: The proportion of true results (both true positives and true negatives) among the total number of cases examined.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Accuracy measures the overall correctness of the model's predictions.

Precision:

$$\text{Precision} = \frac{TP}{TP+FP}$$

Precision measures the proportion of true positive predictions out of all positive predictions made by the model.

Recall (Sensitivity or True Positive Rate):

$$\text{Recall} = \frac{TP}{TP+FN}$$

Recall measures the proportion of true positives that were correctly identified by the model.

Specificity (True Negative Rate):

$$\text{Specificity} = \frac{TN}{TN+FP}$$

Specificity measures the proportion of true negatives that were correctly identified by the model.

F1-Score:

$$\text{F1-Score} = 2 * \frac{Precision*Recall}{Precision+Recall}$$

The F1-score is the harmonic mean of precision and recall and provides a balanced assessment of the model's performance.

Each cell in the confusion matrix corresponds to different classification outcomes, and these formulas allow us to derive important metrics that provide a deeper understanding of how well the model is performing in terms of its predictive accuracy and error types[17].

Let’s consider a detailed example to illustrate the application of the confusion matrix.

Example Scenario: Fraud Detection

Suppose we have developed a model to detect fraudulent transactions in a financial system. We test the model on a dataset of 10,000 transactions, with the following confusion matrix results:

From this matrix:

- True Positives (TP) = 70 (Fraudulent transactions correctly identified)
- False Positives (FP) = 20 (Legitimate transactions incorrectly identified as fraudulent)
- True Negatives (TN) = 9880 (Legitimate transactions correctly identified)
- False Negatives (FN) = 30 (Fraudulent transactions missed)

We can now calculate the various performance metrics:

Accuracy=$\frac{TP+TN}{TP+TN+FP+FN}$

Precision=$\frac{TP}{TP+FP}$

Table 1: The hypothetical scores and calculations of our proposed model.

Threshold	TP	FP	TN	FN	TPR	FPR
0.0	100	900	0	0	1.00	1.00
0.1	95	200	700	5	0.95	0.22
0.2	90	100	800	10	0.90	0.11
0.3	85	50	850	15	0.85	0.06
0.4	80	20	880	20	0.80	0.02
0.5	75	10	890	25	0.75	0.01
0.6	70	5	895	30	0.70	0.01
0.7	65	3	897	35	0.65	0.003
0.8	60	1	899	40	0.60	0.001
0.9	50	0	900	50	0.50	0.00
1.0	0	0	900	100	0.00	0.00

Statistical Analysis:

The project on IDS (Intrusion Detection System) comprises of various statistical point of view through which we can easily demonstrate the working of our Project on IDS. Using WEKA, we trained our dataset(80percent) and rest 20 percent for testing. After training and testing we found accuracy to be approx. 99%. In this process we found various statistical point of view, here some of them are listed below:

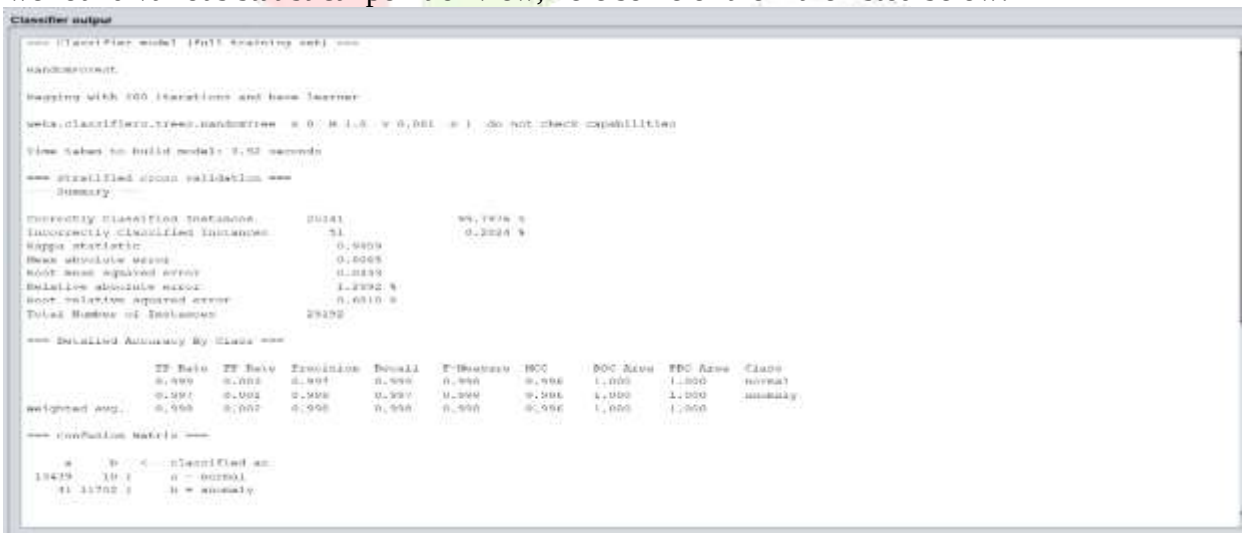

Figure 9: Random forest summary using WEKA

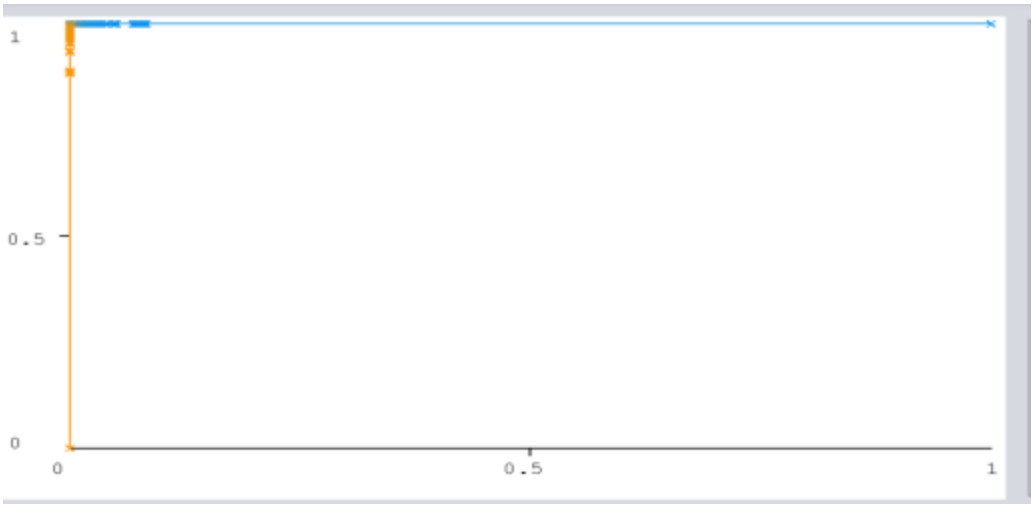

Figure 10: Plot (Area under ROC =1)

Figure 11: Threshold Curve (anomaly)

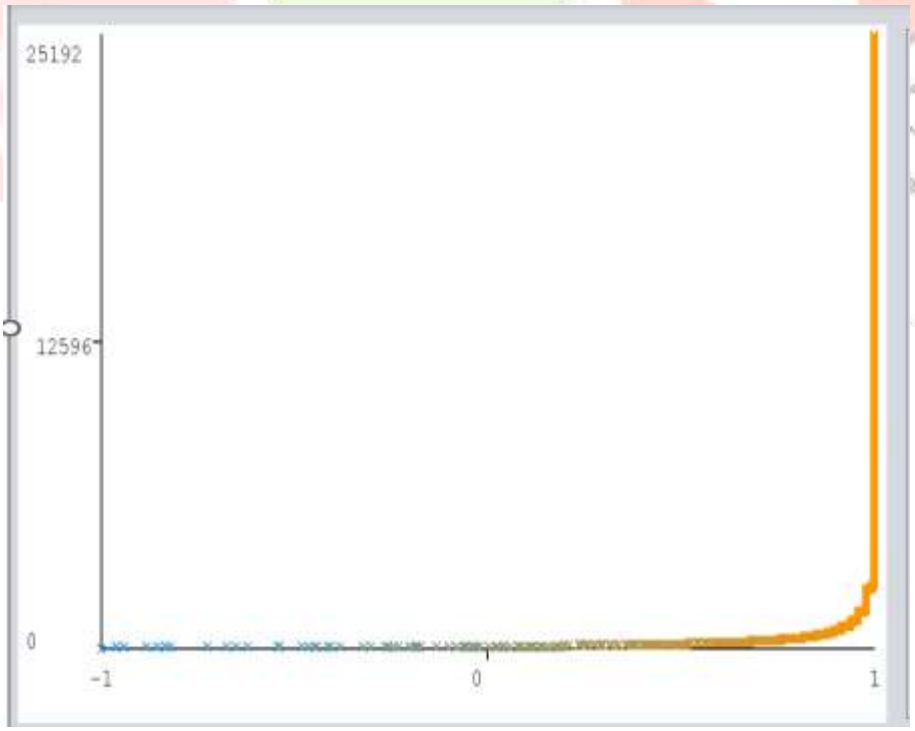

Figure 12:Visual Threshold Curve Anomaly

Conclusion and Future work

In an era where computer networks underpin virtually all aspects of modern life, their security has become a paramount concern. Consequently, robust countermeasures such as encryption, regular software updates, network segmentation, and user education are critical in mitigating these risks. Among these defences, the Intrusion Detection System (IDS) emerges as a pivotal component, providing real-time surveillance and analysis of network activities to detect and thwart malicious actions. By delving into the NSL-KDD dataset, we have explored the efficacy of various classification algorithms in identifying different types of cyberattacks, namely DoS, Probe, U2R, and R2L. The comprehensive analysis has yielded insights into the performance of these algorithms, highlighting their strengths and computational efficiencies. This meticulous approach ensures that IDS can be fine-tuned to enhance detection capabilities, thereby fortifying network defences. The findings of this research underscore the critical need for a multi-layered defence strategy in cybersecurity. Furthermore, the continuous evolution of cyber threats necessitates an adaptive and proactive security posture, integrating advanced machine learning techniques and real-time analytics to stay ahead of potential attacks. Ultimately, the integration of a robust IDS within network infrastructures not only bolsters security but also ensures compliance with regulatory standards, enhances user trust, and preserves the fundamental integrity of digital communications. As cyber threats continue to evolve, the ongoing refinement of IDS and other security mechanisms will be essential in safeguarding the digital backbone of our globalized society. This study contributes to the broader understanding of intrusion detection systems, providing a foundation for future research and development aimed at enhancing network security in an increasingly interconnected world. Through continuous vigilance and innovative security solutions, we can protect the invaluable data and communication networks that drive progress and connectivity in the modern era.

In future our aim is to apply some hybrid deep learning technique to get better accuracy and provides the novel approach for IDS.

References:

1. Safi Ullah, Jawad Ahmad, Muazzam A. Khan, Mohammed S. Alshehri, Wadii Boulila, Anis Koubaa, Sana Ullah Jan, M Munawwar Iqbal Ch, TNN-IDS: Transformer neural network-based intrusion detection system for MQTT-enabled IoT Networks, Computer Networks, Volume 237,2023,110072, ISS13891286, doi.org/10.1016/j.comnet.2023.110072.

2. Tarek Gaber, Joseph Bamidele Awotunde, Mohamed Torky, Sunday A. Ajagbe, Mohammad Hammoudeh, Wei Li,Metaverse-IDS: Deep learning-based intrusion detection system for Metaverse-IoT networks, InternetofThings, Volume24, 2023, 100977, ISSN2542-6605, doi.org/10.1016/j.iot.2023.100977.

3. Shahid Latif, Wadii Boulila, Anis Koubaa, Zhuo Zou, Jawad Ahmad, DTL-IDS: An optimized Intrusion Detection Framework using Deep Transfer Learning and Genetic Algorithm, Journal of Network and Computer Applications, Volume 221, 2024, 103784, ISSN 1084-8045, doi.org/10.1016/j.jnca.2023.103784.
4. Alaa Firas Jasim Jasim, Sefer Kurnaz, New automatic (IDS) in IoTs with artificial intelligence technique, Optik, Volume 273, 2023, 170417, ISSN 00304026, doi.org/10.1016/j.ijleo.2022.170417.

5.Farhan Ullah, Shamsher Ullah, Gautam Srivastava, Jerry Chun-Wei Lin, IDS-INT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic, Digital Communications andNetworks,2023, ISSN 2352-8648, doi.org/10.1016/j.dcan.2023.03.008.

6. Gorby Kabasele Ndonda, Ramin Sadre, Network trace generation for flow-based IDS evaluation in control and automation systems, International Journal of Critical Infrastructure Protection, Volume 31, 2020, 100385, ISSN 1874-5482, doi.org/10.1016/j.ijcip.2020.100385.

7. Sergio Iglesias Pérez, Santiago Moral-Rubio, Regino Criado, A new approach to combine multiplex networks and time series attributes: Building intrusion detection systems (IDS) in cybersecurity, Chaos, Solitons & Fractals, Volume 150, 2021, 111143, ISSN 0960-0779, doi.org/10.1016/j.chaos.2021.111143.

8. Komal Singh Gill, Sharad Saxena, Anju Sharma, GTM-CSec: Game theoretic model for cloud security based on IDS and honeypot, Computers & Security, Volume 92, 2020, 101732, ISSN 0167-4048, doi.org/10.1016/j.cose.2020.101732.

9. Mary McCarron, Darren McCausland, Eimear McGlinchey, Sarah Bowman, Michael Foley, Margaret Haigh, Eilish Burke, Philip McCallion, Recruitment and retention in longitudinal studies of people with intellectual disability: A case study of the Intellectual Disability Supplement to the Irish Longitudinal Study on Ageing (IDS-TILDA), Research in Developmental Disabilities, Volume 124, 2022, 104197, ISSN 0891-4222, doi.org/10.1016/j.ridd.2022.104197.

10. Lorena Cazorla, Cristina Alcaraz, Javier Lopez, A three-stage analysis of IDS for critical infrastructures, Computers & Security, Volume 55, 2015, Pages 235-250, ISSN 0167-4048, doi.org/10.1016/j.cose.2015.07.005.
11. P.M. Mafra, J.S. Fraga, A.O. Santin, Algorithms for a distributed IDS in MANETs, Journal of Computer and System Sciences, Volume 80, Issue 3, 2014, Pages 554-570, ISSN 0022-0000, doi.org/10.1016/j.jcss.2013.06.011.

12. Zouhair Chiba, Noreddine Abghour, Khalid Moussaid, Amina El omri, Mohamed Rida, Intelligent approach to build a Deep Neural Network based IDS for cloud environment using combination of machine learning algorithms, Computers & Security, Volume 86, 2019, Pages 291-317, ISSN 0167-4048, doi.org/10.1016/j.cose.2019.06.013.

13. Xavier Clotet, José Moyano, Gladys León, A real-time anomaly-based IDS for cyber-attack detection at the industrial process level of Critical Infrastructures, International Journal of Critical Infrastructure Protection, Volume 23, 2018, Pages 11-20, ISSN 1874-5482, doi.org/10.1016/j.ijcip.2018.08.002.

14. Xiao-Han Wang, Bingyou Jiang, Chunshan Zheng, Gaochao Pan, Shiju Wang, Yi Zhang, Ben Ji,Influence of IDS and SDS compound ratio on pore structure and wettability of bituminous coal and anthracite: Experimental and simulation discussion, Journal of Molecular Liquids, Volume 406,2024,125037,ISSN 0167-7322,https://doi.org/10.1016/j.molliq.2024.125037.

15. Mattia Giovanni Spina, Mauro Tropea, Floriano De Rango, SURA-LB: Software-defined IDS with UAV Resource Aware Load-Balancing in FANET disaster scenarios,Computer Communications,Volume 223,2024, Pages 101-114, ISSN 0140-3664, https://doi.org/10.1016/j.comcom.2024.05.015.

16. Zhiyu Ma, Chen Li, Tianming Du, Le Zhang, Dechao Tang, Deguo Ma, Shanchuan Huang, Yan Liu, Yihao Sun, Zhihao Chen, Jin Yuan, Qianqing Nie, Marcin Grzegorzek, Hongzan Sun,AATCT-IDS: A benchmark Abdominal Adipose Tissue CT Image Dataset for image denoising, semantic segmentation, and radiomics evaluation,Computers in Biology and Medicine,Volume 177,2024, 108628, ISSN 0010-4825, https://doi.org/10.1016/j.compbiomed.2024.108628.

17. Deguo Ma, Chen Li, Tianming Du, Lin Qiao, Dechao Tang, Zhiyu Ma, Liyu Shi, Guotao Lu, Qingtao Meng, Zhihao Chen, Marcin Grzegorzek, Hongzan Sun, PHE-SICH-CT-IDS: A benchmark CT image dataset for evaluation semantic segmentation, object detection and radiomic feature extraction of perihematomal edema in spontaneous intracerebral hemorrhage,Computers in Biology and Medicine,Volume 173,2024,108342, ISSN 0010-4825, https://doi.org/10.1016/j.compbiomed.2024.108342.

18. Samira Nahim-Granados, Ilaria Berruti, Isabel Oller, María Inmaculada Polo-López, Sixto Malato, Assessment of a commercial biodegradable iron fertilizer (Fe3+-IDS) for water treatment by solar photo-Fenton at near-neutral pH, Catalysis Today,Volume 434,2024,114699,ISSN 0920-5861, https://doi.org/10.1016/j.cattod.2024.114699.

19. Umer Zukaib, Xiaohui Cui, Chengliang Zheng, Dong Liang, Salah Ud Din, Meta-Fed IDS: Meta-Learning and Federated Learning Based Fog-Cloud Approach to Detect Known and Zero-Day Cyber Attacks in IoMT Networks, Journal of Parallel and Distributed Computing,2024,104934,ISSN 0743-7315,https://doi.org/10.1016/j.jpdc.2024.104934.

20. Dechao Tang, Chen Li, Tianmin Du, Huiyan Jiang, Deguo Ma, Zhiyu Ma, Marcin Grzegorzek, Tao Jiang, Hongzan Sun,ECPC-IDS: A benchmark endometrial cancer PET/CT image dataset for evaluation of semantic segmentation and detection of hypermetabolic regions,Computers in Biology and Medicine, Volume 171,2024,108217,ISSN 0010-4825, https://doi.org/10.1016/j.compbiomed.2024.108217.