Computer Fraud and Security
ISSN (online): 1873-7056

A Hybrid Approach for Feature Extraction of Multi-Class Dataset in IDS

Jhansi Rani Mettu[1], Dr Dhanpratap Singh[2]

[1]*School of Computer Science Engineering, Lovely Professional University Phagwara Punjab, India.*

E-mail: jhansirani512@gmail.com

[2]*School of Computer Science Engineering, Lovely Professional University Phagwara Punjab, India.*

E-mail: dhanpratap.25706@lpu.co.in

Abstract:

Intrusion Detection Systems (IDS) are very important for keeping networks safe from online dangers, especially when there are a lot of different classes of data and duplicate features that can slow things down. This work presents a new way to improve the performance of IDS by mixing the K-Best and Random Forest Importance methods for feature extraction. Before Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Independent Component Analysis (ICA), and the suggested blend method were used on the IoT-23 dataset, features were normalized and labelled. The combination method picked out important factors like flow time, packet length, protocol type, and response numbers, which led to better classification results. We used an 80-20 split for train and test to check how well three models (XGBoost, Random Forest, and Naive Bayes) worked. Comparative research showed that the combined method was better, as it achieved 99% accuracy and big gains in precision, memory, and F1-score measures. In particular, XGBoost proved to be the best model, showing impressive speed with its mixed feature set. PCA, LDA, and ICA, on the other hand, gave average results. This shows how important it is to combine different feature selection methods. The results show that the mix method can deal with feature duplication and improve IDS performance, which makes it a good choice for real-world use. To make this method even better, more study could look into how it works with bigger datasets and more models.

Keywords: Intrusion Detection System (IDS), Hybrid Feature Extraction, K-Best, Random Forest Importance, Multi-Class Classification

1. Introduction

Intrusion Detection Systems (IDS) are crucial for defending digital systems from numerous cyber threats, particularly as devices and networks become increasingly interconnected. Since IoT and other data-heavy settings are becoming more popular, attacks are tougher to discover and stop. These networks contain a lot of unique data, thus they require robust analytic techniques to find issues rapidly. IDSs perform better or worse depending on the quality and usefulness of their threat detection capabilities. High-dimensional datasets sometimes contain repeated or forgotten trends, making system learning models less helpful [1]. Thus, strong feature extraction approaches are required to improve IDS and lower computation costs. Many people have employed typical function extraction methods like PCA, LDA, and ICA to improve type and reduce dimensions. These strategies work in certain cases but fail on multi-class datasets with intricate developments. Those issues demonstrate the need of using mixed methods that combine the best of two methods to pick excellent capabilities. Combining approaches [2] may eliminate redundancies, improve the way critical capabilities are displayed, and make IDSs better at spotting issues. This study offers combining okay-pleasant with Random forest importance to extract capabilities, which may detect relevant traits. Random forest significance is dependent on how much they increase version performance, whereas okay-first-rate selects great features based on statistics scores. Combining these strategies ensures that all trends are statistically significant and valuable in real life. This reduces noise in the dataset and improves the chosen functions for distinguishing instructions, especially when there are many [3]. The proposed method was tested on the IoT-23 dataset, which contains all IoT-related network traffic data. Normalization and label encoding ensured data consistency for system learning. Flow duration, protocol type, and response counts were found using hybrid feature extraction. We next trained three popular classifiers—XGBoost, Random Forest, and Naive Bayes—with these attributes. The combined strategy outperformed other feature extraction strategies in model performance. The findings showed that multimodal feature extraction outperformed PCA, LDA, and ICA. The combination technique with XGBoost achieved 99% accuracy and large increases in accuracy, recall,

and F1-score. Traditional approaches failed, demonstrating the value of K-Best and Random Forest Importance. This illustrates that the mixed technique solves IDS application challenges with high-dimensional, multi-class datasets.

2. Related Work

Feature extraction may be crucial to building Intrusion Detection Systems (IDS), especially for multi-class datasets that are so unique and sophisticated. There is study on how to make IDS highlight extraction more successful. Principal Component Analysis (PCA) has been used for a long time since it reduces the number of measurements by splitting data into diverse sections. For instance, [4] experts showed how it may detect high-impact discrepancies. Linear Discriminant Analysis (LDA) [5] uses highlights to distinguish classes and may be used with supervision. The authors of [6] employed LDA to improve IDS classification, however they found that it didn't function well on non-linear datasets. ICA, or Independent Component Analysis, may uncover statistically distinct patterns in noisy datasets [7]. It takes too much computer resources and is sensitive to scale to be utilized in real time [8]. To avoid these issues, hybrid approaches are prevalent. Some research [9, 10] advised combining PCA with LDA for best results. These hybrid techniques showed potential, but linear changes prevent them from capturing multi-class datasets' complicated feature relationships. Adding machine learning-based methodologies such ensemble feature value measurement has richened feature extraction. Decision tree-friendly characteristics are rated by Random Forest Importance. This makes selecting the most significant characteristics in multidimensional datasets straightforward [11]. Research in [12] found that combining Random Forest Importance with statistical approaches improved categorization. Because it is simple and effective, K-Best, a statistical approach that ranks features by target variable correlation, is also extensively employed [13]. As demonstrated in [14], standalone K-Best algorithms may not capture non-linear interactions.

New advances show why we need hybrid statistics and machine learning methodologies. A combination of K-Best and Mutual Information was proposed in [15] to discover anomalies more efficiently. Similar to [16], researchers chose IoT data attributes using a combination model of Chi-Square and Random Forest Importance. These strategies demonstrated that combining the best components of many techniques might compensate for their drawbacks. The IoT-23 dataset includes a lot of IoT-related network traffic, therefore it is used to evaluate IDS feature extraction algorithms [17]. The dataset was used to evaluate conventional and hybrid feature extraction algorithms [18]. They discovered hybrid techniques often outperformed solo ones. Before hybrid techniques may be trusted, feature normalization and encoding are necessary [19]. In [20] and [21], hybrid feature extraction approaches and sophisticated algorithms like XGBoost have substantially improved recognition. Even with these gains, computers struggle to balance speed and classification. We need to learn how to employ mixed approaches with larger datasets and adapt them to new threats.

3. Used CIC IoT dataset 2023

The Canadian Institute for Cybersecurity created the CIC IoT Dataset 2023 as a complete tool to help with study in attack detection and IoT security. This dataset has a lot of information about network activity that is suited to the changing security needs of IoT settings. It records both legitimate and illegal data from many Internet of Things (IoT) devices in the real world, giving us a fair sample for teaching and testing machine learning models. The dataset has many different kinds of traits that were taken from network traffic. These include information about individual packets, flow factors, and protocol-specific parameters. Some important factors are the length of the flow, the size of the packets, the inter-arrival time (IAT), and flags like SYN, ACK, and FIN numbers.

	flow_duration	Header_Length	Protocol Type	Duration	Rate	Srate	Drate	fin_flag_number
0	0.000000	54.0	6.0	64.0	8.216014	8.216014	0.0	0.0
1	0.000000	0.0	1.0	64.0	1.273160	1.273160	0.0	0.0
2	0.197337	234065.0	17.0	64.0	2139.548403	2139.548403	0.0	0.0
3	0.107838	30854.5	17.0	64.0	11434.330849	11434.330849	0.0	0.0
4	4.636251	108.0	6.0	64.0	0.431384	0.431384	0.0	0.0

Figure 1: Sample Dataset Description

There are also tools that are special to TCP, UDP, and ICMP data, which lets you analyze different types of communication in IoT systems in great detail. The collection is notable for its big size it includes millions of records that model how IoT devices would talk to each other during different types of attacks, such as DoS, DDoS, and data theft. It's careful marking of regular and attack traffic makes guided learning tasks very reliable. Time-stamped data and information give the dataset historical context, which makes it very useful for time-series analysis and real-time recognition systems. Because of this, it is an important tool for improving IDS study in IoT security.

3. Methodology

The suggest proposed method, flowchart illustrate in figure 2, mixed feature extraction method that combines K-Best and Random Forest Importance techniques, the suggested method aims to improve intrusion detection in multi-class datasets. In experiments, the IoT-23 dataset is used because it contains a lot of real-world IoT network information, both good and bad. As the first step in the process, the raw information is normalized to make the feature values more consistent and label encoding is used to turn category values into numbers. This gets rid of data errors and makes sure it works with machine learning models. In order to set the input factors and goal groups for classification tasks, features and labels are kept separate. After that, hybrid feature extraction is used to make feature selection better and reduce the number of dimensions. The K-Best method starts by ranking features based on how statistically important they are to the goal variable. This finds the most important factors. Next, Random Forest Importance checks the dataset by figuring out how important each feature is for building decision trees, with the focus being on the features that are most important for classification performance. By using these methods together, you can be sure that both statistically significant and powerful features will be kept. There is an 80/20 split between the training set and the testing set in the revised dataset. XGBoost, Naive Bayes, and Random Forest are some of the algorithms that are used to test how well the mixed feature set works. To show that the suggested method is better than standard feature extraction methods, performance measures like accuracy, precision, recall, and F1-score are measured and compared.

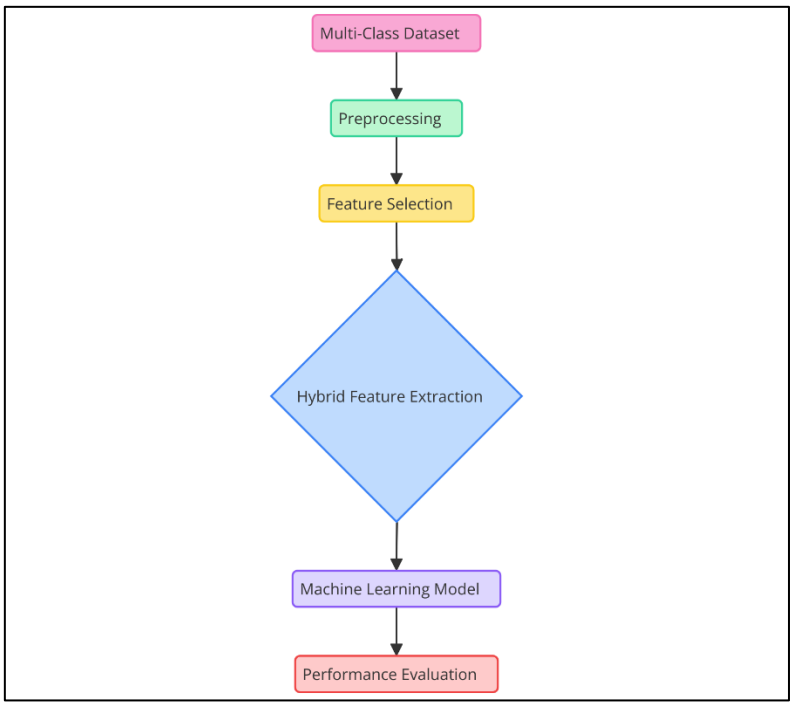

Figure 2: Overview of workflow for proposed system architecture

A. Data preprocessing

1. Features and Labels Separation

During preprocessing, the information is split into features, which are the input factors, and labels, which are the desired outcomes. Features are the distinctive parts of arrange information, like bundle estimate, stream length, and convention sort. They are exceptionally imperative for finding patterns or interesting behavior. On the other hand, names appear whether the information is related to typical behavior or a certain kind of assault. By isolating highlights from names, machine learning calculations can center on the interface between inputs and yields. This step makes beyond any doubt that forecasts and target bunches are clearly isolated, which makes building a solid demonstrate less demanding.

Let the dataset `D` consist of `n` samples with `m` features:

$$D = \{ (X1, y1), (X2, y2), \ldots, (Xn, yn) \}$$

Where:

- Xi = [xi1, xi2, ..., xim]: Feature vector of the i-th sample.

- yi: Label corresponding to Xi.

The separation process splits `D` into:

$$- Feature\ matrix\ `X`: X = [X1, X2, \ldots, Xn] \in R\hat{}(n \times m).$$

$$- Label\ vector\ `Y`: Y = [y1, y2, \ldots, yn] \in R\hat{}n.$$

2. Label Encoder

Label decoding may be a exceptionally imperative step for turning names that portray categories into numbers that machine learning frameworks can get it. Within the IoT-23 dataset, for occurrence, names that appear the sort of activity (such as "Ordinary," "DoS," or "DDoS") are turned into numbers (0, 1, 2). This encoding keeps the data's structure and lets computers utilize math to figure out what category factors cruel. It works particularly well when there are more than two classes to sort, keeping the associations between the names intaglio. This step plans the dataset for simple utilize in machine learning forms by carefully putting away names.

Let `L = {l1, l2, ..., lk}` represent `k` unique categorical labels. Label encoding maps these labels to integers:

$$f(lj) = j - 1, \forall\, lj \in L, j = 1, 2, \ldots, k$$

Thus, each label `yi ∈ L` is converted into `f(yi) ∈ {0, 1, ..., k-1}`.

Example:

$$-If\ L = \{"Normal", "DoS", "DDoS"\},$$

$$f("Normal") = 0, f("DoS") = 1, f("DDoS") = 2.$$

3. Features Normalization

Normalizing features gets rid of scale differences between parameters by making the range of numerical features the same. Values like file size and inter-arrival time, for example, can be very different. Most of the time, normalization changes the range of these numbers to be between 0 and 1. This step makes sure that the model treats all factors similarly by stopping features with larger magnitudes from controlling the learning process. Normalization also speeds up the completion of optimization methods while the model is being trained, which makes it work faster and more accurately.

Normalization rescales each feature `xij` to a standard range (e.g., [0, 1]):

$$x'ij = \frac{xij - \min(xj)}{\max(xj) - \min(xj)}, \forall\, i \in [1, n], j \in [1, m]$$

Where:

- xij: Original value of the j-th feature in the i-th sample.

- min(xj): Minimum value of feature `j` across all samples.

- max(xj): Maximum value of feature `j` across all samples.

This ensures that x'ij ∈ [0, 1], making the dataset uniform and reducing biases caused by varying feature scales.

4. Result and Discussion

XGBoost, Naive Bayes (NB), and Random Forest (RF) classifiers' performance was tested using various feature extraction methods, as shown in Table 1: PCA, LDA, ICA, and a combination method that combines K-Best and Random Forest Importance of Features. Crucial measures like accuracy, precision, memory, and F1-score were used to rate each method.

Table 3: Performance Comparison of Algorithms with various feature extraction technique

Method	**Accuracy**	**Precision**	**Recall**	**F1-Score**
PCA Features Extraction				
XGBoost	93	62	60	60
NB	75	39	44	37
RF	94	62	60	60
LDA Features Extraction				
XGBoost	93	64	58	59
NB	72	40	40	36
RF	94	62	56	58
ICA Features Extraction				
XGBoost	95	62	60	60
NB	73	39	44	35
RF	95	64	61	64
Hybrid Feature Extraction (K-BEST + Random Forest Importance)				
XGBoost	**99**	**77**	**70**	**71**

NB	71	45	44	36
RF	99	70	68	69

The figure 3 shows how well different feature extraction methods work with different algorithms (XGBoost and Random Forest). The combination method (K-Best + Random Forest Importance) had the best accuracy (99%) and F1-score (71% for XGBoost and 69% for RF), showing that it was the best at improving classification results. Other methods, such as PCA, LDA, and ICA, gave average results, with ICA slightly doing better than LDA and PCA. XGBoost always did better than Random Forest in all methods for extracting features.

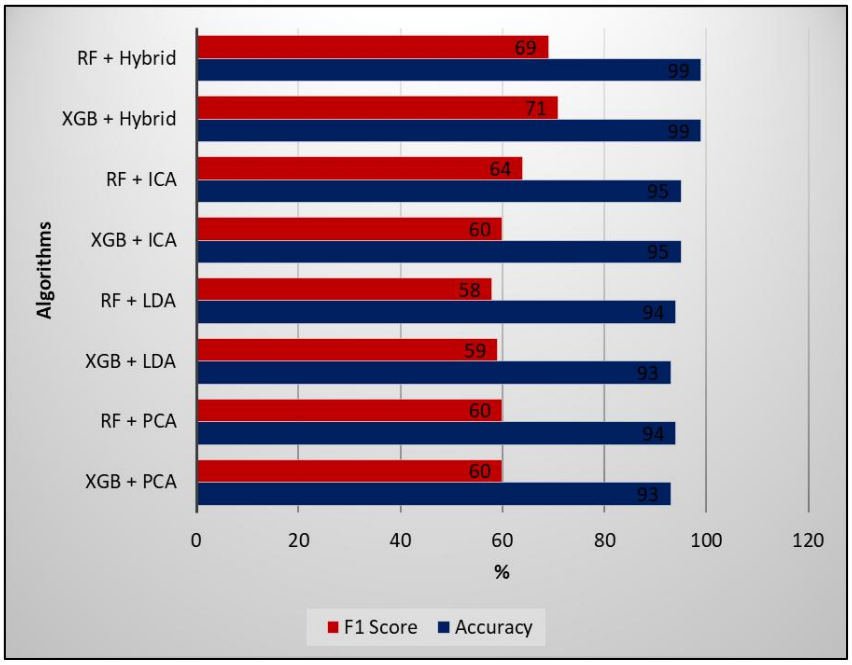

Figure 3: Performance Comparison Graph of Algorithm with Feature Extraction Techniques

The accuracy, precision, recall, and F1-score of Naive Bayes (NB), Random Forest (RF), and XGBoost are shown and compared in Figure 4. At 99%, both XGBoost and RF are the most accurate. However, XGBoost is better than RF in precision (77% vs. 70%) and F1-score (71% vs. 69%). NB is much behind, with an accuracy score of 71%, a precision score of 45%, and an F1-score of 36%. The outcomes show that XGBoost is better at dealing with large datasets, especially when mixed with hybrid feature extraction methods. This makes it the best algorithm for finding intrusions.

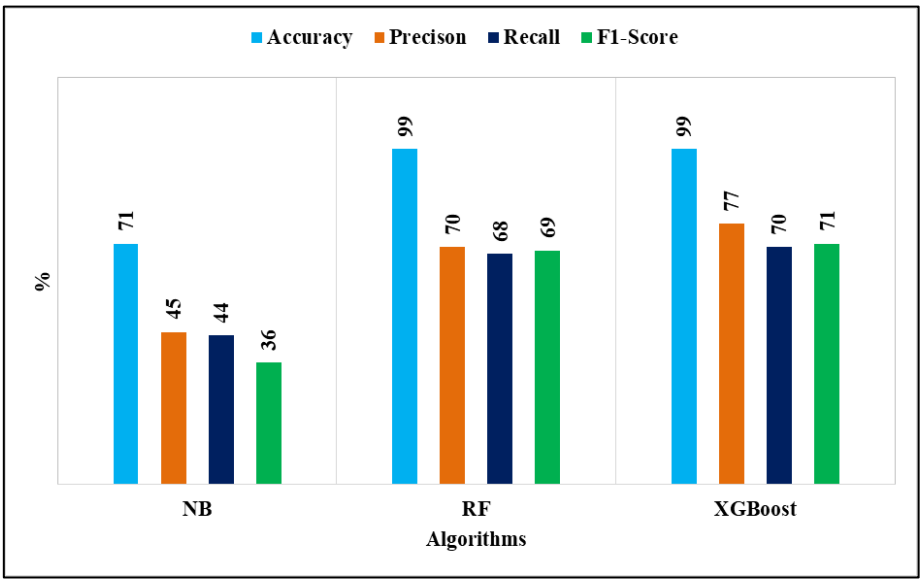

Figure 4: Performance Comparison of Algorithms WRT Hybrid Feature Extraction Technique

Combining the best features of K-Best and Random Forest Importance to make feature selection better, the combination method was the clear winner. Because of its gradient optimization and ability to handle feature interactions, XGBoost regularly did better than NB and RF as a classifier. A method using both hybrid feature extraction and XGBoost works best for making intrusion detection systems very accurate and reliable, especially for datasets with multiple classes like IoT-23.

5. Conclusion

In multi-class classification situations, where feature variety and duplication are big problems, intrusion detection systems (IDS) are very important for keeping networks safe. The IoT-23 dataset was used in this study to show how to use a mixed feature extraction method that combines K-Best and Random Forest Importance to get the best feature selection for IDS. When we compared different classifiers (XGBoost, Random Forest, and Naive Bayes) and feature extraction methods (PCA, LDA, ICA, and Hybrid), it was clear that the hybrid method was the best at making classification work better. The combination method did a great job because it kept important things like flow length, protocol type, and response numbers. With a score of 71% on the F1 scale, XGBoost was the best predictor, with 99% accuracy, 77% precision, and a mixed feature set. Random Forest was right behind it, with 99% accuracy and an F1-score of 69%, which shows how solid it is. On the other hand, Naive Bayes didn't do very well. Its best accuracy score was 71%, and its F1-score was only 36%. This shows that it doesn't work well with large, multi-class datasets. Traditional feature extraction methods, such as PCA, LDA, and ICA, made some progress but had trouble dealing with the complexity of links between more than one classes. ICA did better than the other ways, but it couldn't be used in real time because it needed a lot of computing power. Using statistics and machine-learning-based methods, the mixed approach was able to solve these problems by making features much more relevant and lowering their duplication. When used together, the hybrid feature extraction method and XGBoost show great promise for intrusion detection systems (IDS) because they provide a strong and flexible way to find attacks in complicated, multi-class network settings. This work could be expanded in the future by adding more datasets and testing mixed methods with deep learning models to make recognition even better and more scalable.

References

[1] Ahmadi Abkenari, F.; Milani Fard, A.; Khanchi, S. Hybrid Machine Learning-Based Approaches for Feature and Overfitting Reduction to Model Intrusion Patterns. J. Cybersecur. Priv. 2023, 3, 544-557. https://doi.org/10.3390/jcp3030026

[2] Le, K.-H.; Nguyen, M.-H.; Tran, T.-D.; Tran, N.-D. IMIDS: An Intelligent Intrusion Detection System against Cyber Threats in IoT. Electronics 2022, 11, 524.

[3] Joo, H.; Choi, H.; Yun, C.; Cheon, M. Efficient Network Traffic Classification and Visualizing Abnormal Part Via Hybrid Deep Learning Approach: Xception + Bidirectional GRU. Glob. J. Comput. Sci. Technol. 2022, 21, 1–10.

[4] Hindy, H.; Bayne, E.; Bures, M.; Atkinson, R.; Tachtatzis, C.; Bellekens, X. Machine Learning Based IoT Intrusion Detection System: An MQTT Case Study (MQTT-IoT-IDS2020 Dataset). Lect. Notes Netw. Syst. 2021, 180, 73–84.

[5] Farooq, M.S.; Abbas, S.; Rahman, A.U.; Sultan, K.; Khan, M.A.; Mosavi, A. A fused machine learning approach for intrusion detection system. Comput. Mater. Contin. 2023, 74, 2607–2623.

[6] Kocher, G.; Kumar, G. Machine learning and deep learning methods for intrusion detection systems: Recent developments and challenges. Soft Comput. 2021, 25, 9731–9763.

[7] Aversano, L.; Bernardi, M.L.; Cimitile, M.; Pecori, R. A systematic review on Deep Learning approaches for IoT security. Comput. Sci. Rev. 2021, 40, 100389.

[8] Henry, A.; Gautam, S.; Khanna, S.; Rabie, K.; Shongwe, T.; Bhattacharya, P.; Sharma, B.; Chowdhury, S. Composition of Hybrid Deep Learning Model and Feature Optimization for Intrusion Detection System. Sensors 2023, 23, 890. https://doi.org/10.3390/s23020890

[9] R. N. Wadibhasme, A. U. Chaudhari, P. Khobragade, H. D. Mehta, R. Agrawal and C. Dhule, "Detection And Prevention of Malicious Activities In Vulnerable Network Security Using Deep Learning," 2024 International Conference on Innovations and Challenges in Emerging Technologies (ICICET), Nagpur, India, 2024, pp. 1-6, doi: 10.1109/ICICET59348.2024.10616289.

[10] Alhaidari, F.; Abu-Shaib, N.; Alsafi, M.; Alharbi, H.; Alawami, M.; Aljindan, R.; Rahman, A.-U.; Zagrouba, R. ZeVigilante: Detecting Zero-Day Malware Using Machine Learning and Sandboxing Analysis Techniques. Comput. Intell. Neurosci. 2022, 2022, 1615528.

[11] Alqarni, A.; Rahman, A. Arabic Tweets-Based Sentiment Analysis to Investigate the Impact of COVID-19 in KSA: A Deep Learning Approach. Big Data Cogn. Comput. 2023, 7, 16.

[12] Alotaibi, A.; Rahman, A.; Alhaza, R.; Alkhalifa, W.; Alhajjaj, N.; Alharthi, A.; Abushoumi, D.; Alqahtani, M.; Alkhulaifi, D. Spam and sentiment detection in Arabic tweets using MARBERT model. Math. Model. Eng. Probl. 2022, 9, 1574–1582.

[13] Basheer Ahmed, M.I.; Zaghdoud, R.; Ahmed, M.S.; Sendi, R.; Alsharif, S.; Alabdulkarim, J.; Albin Saad, B.A.; Alsabt, R.; Rahman, A.; Krishnasamy, G. A Real-Time Computer Vision Based Approach to Detection and Classification of Traffic Incidents. Big Data Cogn. Comput. 2023, 7, 22.

[14] Alghamdi, A.S.; Rahman, A. Data Mining Approach to Predict Success of Secondary School Students: A Saudi Arabian Case Study. Educ. Sci. 2023, 13, 293.

[15] Musleh, D.; Alotaibi, M.; Alhaidari, F.; Rahman, A.; Mohammad, R.M. Intrusion Detection System Using Feature Extraction with Machine Learning Algorithms in IoT. J. Sens. Actuator Netw. 2023, 12, 29. https://doi.org/10.3390/jsan12020029

[16] Megantara, A.A.; Ahmad, T. A hybrid machine learning method for increasing the performance of Network Intrusion Detection Systems. J. Big Data 2021, 8, 142.

[17] De Carvalho Bertoli, G.; Pereira Junior, L.A.; Saotome, O.; Dos Santos, A.L.; Verri, F.A.; Marcondes, C.A.; Barbieri, S.; Rodrigues, M.S.; Parente De Oliveira, J.M. An end-to-end framework for machine learning-based network Intrusion Detection System. IEEE Access 2021, 9, 106790–106805.

[18] Wang, M.; Zheng, K.; Yang, Y.; Wang, X. An explainable machine learning framework for Intrusion Detection Systems. IEEE Access 2020, 8, 73127–73141.

[19] Ho, S.; Jufout SAl Dajani, K.; Mozumdar, M. A Novel Intrusion Detection Model for Detecting Known and Innovative Cyberattacks Using Convolutional Neural Network. IEEE Open J Comput Soc. 2021, 2, 14–25.

[20] Priyanka, V.; Gireesh Kumar, T. Performance Assessment of IDS Based on CICIDS-2017 Dataset. In Information and Communication Technology for Competitive Strategies (ICTCS 2020); Lecture Notes in Networks and Systems; Joshi, A., Mahmud, M., Ragel, R.G., Thakur, N.V., Eds.; Springer: Singapore, 2022; Volume 191.

[21] Sun, P.; Liu, P.; Li, Q.; Liu, C.; Lu, X.; Hao, R.; Chen, J. DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system. Secur. Commun Netw. 2020, 2020, 8890306.