International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Improving Intrusion Detection Performance with Genetic Algorithm-Based Feature Extraction and Ensemble Machine Learning Methods

Gunupusala Satyanarayana[1*], Kaila Shahu Chatrapathi[2]

Submitted: 07/05/2023 **Revised**: 17/07/2023 **Accepted**: 05/08/2023

Abstract: The Internet of Things (IoT) has transformed our world by offering enhanced accessibility, connectivity, and convenience in our daily lives. It facilitates the seamless flow of vast amounts of data among interconnected devices, creating a network that is susceptible to diverse network attacks and intrusions. Developing an efficient IDS (Intrusion Detection System) for IoT networks is a challenging task primarily due to two reasons: the massive amount of aggregated data and the diverse nature of IoT devices. Traditional IDS approaches struggle to handle and analyze this data in real time. Hence, there is a growing demand for advanced IDS techniques that leverage ML or DL methods. This study specifically focuses on intrusion detection in IoT networks, utilizing the UNSW-NB15 dataset. The UNSW-NB15 dataset is a well-known and publicly available dataset that is widely used for evaluating the effectiveness of IDS algorithms. The main purpose of the current work is to enhance the performance of intrusion detection by integrating feature extraction techniques based on genetic algorithms (GA) and ensemble machine learning algorithms (EM's). By leveraging these approaches, the study aims to improve the accuracy and effectiveness of detecting intrusions in IoT networks. Feature extraction is a crucial step in IDS, as it aims to reduce the dimensionality of the dataset while retaining relevant information. Genetic algorithms, known for their optimization capabilities, are employed to search for an optimal subset of features that maximize the discriminatory power of the IDS. To achieve this, a framework is proposed that integrates genetic algorithms with various ensemble ML techniques, including random forests, Extra-Trees, XGBoost, AdaBoost, and stacking. The GA selects a subset of features from the UNSW-NB15 dataset, and the ensemble ML models are trained and evaluated using these selected features and calculate accuracy.

Keywords*: EM's Classifier, GA Feature Selection, UNSW-NB15 dataset, Intrusion detection.*

1. Introduction

The Global Internet Statistics Report highlights a significant increase in the number of active internet users, reaching 4.66 billion, and an exponential growth in data generation, surpassing 2 quintillion bytes per day. This surge in data access from diverse sources has led to the rapid development of hacking tools and techniques. Consequently, data security and privacy have become crucial in safeguarding information against intrusions and hostile attacks. However, traditional intrusion detection systems have struggled to detect and respond to assaults or intrusions promptly and efficiently due to the sheer volume and speed of data.

Given the substantial computational complexities arising from the vast amount of data, sophisticated intelligent techniques and robust technologies are required. IDS plays a crucial role in network security by actively monitoring network traffic to identify threats, attacks, or any suspicious activities. Upon detection, an IDS promptly notifies the appropriate administrator. To effectively manage and classify intrusions or attacks, various machine learning techniques can be utilized [1-3].In the realm of network and information system security, IDS were identified as a significant tool for over two decades [4]. IDS plays a crucial role in safeguarding smart Internet of Things (IoT) devices by effectively handling numerous attacks and monitoring suspicious network traffic [5]. Nevertheless, the application of traditional IDS methods in IoT environments encounters challenges due to the distinctive protocol stacks, standards, as well as architectural constraints present in IoT systems. The current solutions fall short of providing comprehensive protection against all forms of attacks, thereby driving the requirement for new approaches. One such method involves the use of physical hardware applications that employ network probes to transmit encrypted data to a remote server for the purpose of detecting malicious activities. However, implementing such solutions demands significant resources [6].

Developing effective IDS systems to repel hacker intrusions remains a significant challenge. Machine learning algorithms offer a promising avenue for detecting suspicious attacks, wherein these algorithms are trained and applied to identify previously unseen patterns in the detection process. Multiple classification algorithms, including various machine learning techniques, are utilized to identify attacks within a

[1*]*Research Scholar, JNTU Hyderabad, India.*
Email: snarayana.5813@gmail.com
[2]*Professor, Department of CSE, JNTU Hyderabad, India.*
Email: shahujntu@gmail.com.

network. Moreover, techniques for feature reduction could be used to enhance classifier performance and speed of detection [7].

The 1st IDS was introduced in 1980, and over the years, numerous mature IDS products were developed. Nevertheless, many IDSs continue to struggle with a significant false alarm rate, resulting in a large number of alerts triggered by non-threatening situations. This issue raises concerns among security analysts, as it increases the likelihood of genuine malicious attacks being overlooked. As a result, academic researchers are actively working towards the development of IDSs with greater detection rates and lower False Alarm Rates (FAR), aiming to enhance the overall effectiveness of intrusion detection systems. Additionally, current IDSs face the challenge of identifying unknown attacks, as network environments are constantly evolving, giving rise to new attacks and their variants. The detection of unknown attacks is crucial, and enhanced IDSs are employed for this purpose.

In the development of intrusion detection systems (IDSs), researchers are increasingly relying on ML (Machine Learning) approaches. ML, as a subset of AI (Artificial Intelligence), has the ability to extract valuable insights from vast datasets. IDSs that leverage ML techniques exhibit notable detection capabilities and demonstrate satisfactory generalizability in identifying unknown attacks and their variations, provided they have sufficient training data. Additionally, machine learning-based IDSs do not require extensive technical knowledge, making them relatively accessible for development and design purposes [8].Although certain comparative studies have been conducted, comprehensive research in this domain is still lacking. Hence, this research aims to develop IDS for networks by employing enhanced feature selection and classification methodologies. Additionally, computational limitations need to be considered during intrusion detection. The IDS must be capable of detecting all types of attacks while maintaining a high level of intrusion detection accuracy. To achieve this, a comprehensive and enhanced feature dataset must be utilized, enabling the investigation of various performance measures. Various effective feature selection techniques will be investigated. The present study focuses on the combination of feature selection techniques with classification methods to enhance the accuracy of IDSs. However, implementing IDS poses certain challenges. Traditional IDS systems' execution is seen as a challenge because of their architectural limitations, protocol stacks, and standards.

This research proposes a novel approach that addresses the aforementioned challenges, focusing on detecting different forms of attacks and improving the accuracy of intrusion rate identification. The aim is to develop an IDS system that is both fast and accurate in identifying all forms of attacks. To obtain this, a unique combination of a Genetic Algorithm with an Ensemble Machine learning methods (GA-EM's) algorithm is proposed. The IDS system utilizes random forests, Extra-Trees, XGBoost, AdaBoost, and stacking for efficient classification. Additionally, a feature selection approach known as Genetic Algorithms (GA) is applied to enhance the classification accuracy, choosing the most relevant features from the dataset. This approach ensures that the IDS system returns the best result from the global optimum. To assess the performance of the suggested IDS system in identifying malicious attacks, the UNSW-NB15 Dataset is utilized for analysis. Comparative assessments are carried out to measure the efficiency of the suggested IDS system against existing models. Through these evaluations, the aim is to demonstrate the improved categorization performance and efficacy of the suggested IDS system in detecting and mitigating various malicious attacks.

The structure of the work is organized as follows: Section 2 offers an analysis of related studies in the field. In Section 3, an overview of the UNSW-NB15 dataset is provided. Section 4 represents a background on the ensemble machine learning (EM's) methods utilized in the analysis. Section 5 introduces the proposed intrusion detection framework. The experimental setup and a detailed discussion of the results are expressed in Section 6. Finally, Section 7 concludes the work, summarizing the findings and highlighting the contributions made in this research.

2. Related Works

Khammassi C et al.[9], the UNSW-NB15 and KDDCup99 datasets were analysed using the GA ("Genetic Algorithm") and LR ("Logistic Regression") wrapper-based feature selection approach. The investigation employed the "Weka simulation" tool. Multiple simulations were performed, and the outcomes revealed that when the GA-LR approach was combined with the DT classifier, a detection score of 81.42% was achieved. By utilizing a subset of 20 features out of the total 42 available in the "UNSW-NB15" feature space, the approach also yielded a FAR of 6.39%. For the dataset of KDDCup99, employing the GA-LR method along with the DT classifier resulted in a detection score of 92.90% and a FAR of 0.105%with 18 features.

Osanaiye O et al.[10],the authors proposed an innovative approach for detecting DDoS ("Distributed Denial of Service") attacks using a filter-based method. They incorporated multiple filters, including Chi-Square, Information Gain, ReliefF, and Gain Ratio, into their analysis. To assess the effectiveness of their system, they

utilized the NSL-KDD attack detection dataset. The authors used the DT ("Decision Tree") algorithm for classification and trained and validated it using the k-fold cross-validation technique with k set to 10. The experimental findings demonstrated that by utilizing only 13 features out of the total feature space of 42, the DT classifier achieved an impressive detection accuracy score of 99.67%. Additionally, it achieved a low FAR of 0.42 % Nevertheless, it is significant to note that this research did not extensively address the multiclass classification problem associated with the NSL-KDD dataset.

Ambusaidi MA et al.[11]the researchers presented an IDS that integrated a filter-based approach for input reduction. They conducted evaluations using various datasets, including Kyoto 2006, KDDCup99, and NSL-KDD. The authors employed the Flexible Mutual Information (FMI) technique, a non-linear correlation measure, to assess the correlation among different input variables. In their experimentation, the researchers selected the LS-SVM ("Least Square Support Vector Machine") as the classifier. The results obtained demonstrated that when applied to the NSL-KDD dataset with 18 features, the LS-SVM FMI obtained an impressive accuracy of 99.94% and a remarkably low FAR of 0.28%. For the KDD Cup 99 dataset, the LS-SVM FMI achieved an overall accuracy of 78.86 %. In the instance of the Kyoto 2006+ dataset, during the tenth iteration, the "LS-SVM FMI" achieved a FAR rate of 0.43% and a detection rate of 97.80%.

Ingre B et al. [12] the authors implemented a filter-based method to develop an IDS aimed at reducing the number of input attributes (features) necessary for training & testing the model. They employed the DT classifier in combination with a correlation-based input selection method. The NSL-KDD dataset was used for their experiments. After applying the filter to the feature space, a total of 14 features were selected. The authors considered both the multiclass classification scenario, which included all 5 kinds of attacks in the NSL-KDD dataset, and the binary classification setup.The experimental findings showed that the system achieved an accuracy of 90.30 percent for the binary setup and 83.66 percent for the multiclass configuration.

Alazzam, H et al.[13]The researchers utilized the Pigeon-Inspired Optimizer (PIO) algorithm for feature reduction in an intrusion detection system. The PIO algorithm draws inspiration from the flight behavior of white pigeons, which continuously adjust their flight position based on the best bird in the flock [14]. Two variations of PIO were employed: Cosine PIO and Sigmoid PIO. The efficacy of their method was assessed using 3 intrusion detection datasets: NSL-KDD, UNSW-NB15, and KDDCup99. For the KDDCup99 dataset, the "Sigmoid PIO" selected 10 features, while the Cosine PIO selected 7 features. In the case of the NSL-KDD dataset, the Cosine PIO selected 5 features, and Sigmoid PIO selected 18 features. Regarding the UNSW-NB15 dataset, the Sigmoid PIO selected 14 features, as well as Cosine PIO, selected 5 features. In terms of accuracy, the Sigmoid PIO achieved 94.7% for the KDDCup99 dataset, 86.9% for the NSL-KDD dataset, and 91.3% for the UNSW-NB15 dataset. On the other hand, the Cosine PIO obtained an accuracy of 96.0% for the KDDCup99 dataset, 88.3% for the NSL-KDD dataset, and 91.7% for the UNSW-NB15 dataset.

Janarthanan T et al. [15] conducted experiments utilizing the UNSW-NB15 dataset to determine the optimal feature space. They implemented various feature selection algorithms, including the CfsSubsetEval attribute evaluator, Information Gain, Greedy Stepwise, and the Ranker technique, using the Weka tool. Through multiple simulations, the researchers obtained two subsets of features for evaluation. The effectiveness of each subset was assessed using the Kappa Statistic measure. Several classifiers were tested, and the RF ("Random Forest") classifier emerged as the best-performing method overall. An accuracy of 75.6617 percent and a Kappa Score of 0.6891 was obtained for the first subset, which had 8 important characteristics. On the other hand, the accuracy of the second subset, consisting of only five important characteristics, was 81.6175 percent with a Kappa value of 0.7639.

Kumar V et al. [16] focused on developing an IDS validated with the UNSW-NB15 dataset. The 22 significant attributes were chosen using a filter-based feature extraction method that was driven by Information Gain. The proposed IDS used an integrated rule-based system that incorporated multiple Tree-based classifiers for classification. Performance evaluation of the IDS included metrics such as AAc ("Attack Accuracy") on test data, FM ("F-Measure"), and FAR. The IDS achieved a FAR of 2.01 percent, FM of 90 percent, and AAc of 57.01 percent. However, it is important to note that exploring alternative machine learning algorithms beyond Tree-based methods could potentially enhance the results, considering the limitations associated with Tree-based approaches.

Almomani O at el. [17] the researchers presented a feature extraction algorithm that utilized GA ("Genetic Algorithm"), GO ("Grey Wolf Optimization"), FO ("Firefly Optimization"), and PSO ("Particle Swarm Optimization"). These algorithms have been iteratively used in the UNSW-NB15 dataset to identify an optimal subset of features, aiming to enhance attack detection accuracy. Following the experimentation, a feature subset comprising 30 features has been chosen. The J48 Tree-

based model and SVM have been employed for classification, with a focus on metrics like Accuracy, FNR ("False Negative Rate"), FPR, and FM. The J48 model obtained a FM of 90.172%, FPR of 14.950%, and training accuracy of 90.484%. On the other hand, the SVM model attained a FNR of 3.130%, FPR of 15.391%, FM of 89.808%, and training accuracy of 90.119%.

Khan NM et al. [18] conducted feature reduction using the RF method to determine the FI ("Feature Importance") scores of attributes in the dataset of UNSW-NB15. The RF algorithm was employed to select a subset of 11 attributes with high importance. For classification, the researchers considered multiple algorithms, including kNN ("k-Nearest Neighbors"), BME ("Bagging Meta Estimator"), DT ("Decision Tree"), XGBoost, and Random Forest (RF). The performance of these algorithms was evaluated based on FM scores and accuracy on test data. Among these algorithms, the RF algorithm yielded the best findings, achieving an accuracy of 75.56 percent and an F-Measure score of 73.00 percent.

Tama BA et al. [19] the authors presented a two-stage (TS) ensemble model for IDS that combined Rotation Forest & Bagging methods. They also utilized GA, PSO, and ACO ("Ant Colony Optimization") for feature selection. Applying the PSO-GA-ACO approach to the UNSW-NB15 dataset, they identified 19 optimal features. To assess the effectiveness of their approach, a 10-fold cross-validation with the hold-out approach was conducted. The performance metrics considered included accuracy, false positive rate (FPR), sensitivity, and precision. In binary classification, the proposed methodology obtained a sensitivity of 91.30%, a precision of 91.60%, and an accuracy of 91.27 % on the subset of UNSW-NB15.

Zong W et al.[20] suggested an IDS that utilized a TS classifier model on the basis of RF classifier. The IG ("Information Gain") approach has been employed for attribute selection in binary classification, and categorical UNSW-NB15 dataset features were transformed using one-hot encoding. The TS classifier model consisted of two stages: the first stage (IG-TS) focused on detecting minority classes, while the second stage aimed at detecting the majority class. The results obtained from each stage were combined to offer the final prediction.The performance evaluation of the IDS included metrics such as accuracy (AC) and false alarm rate (FAR). After conducting multiple tests, the IG-TS obtained an accuracy of 85.78 percent and a false alarm rate of 15.78 percent.

Belouch M et al. [21] the authors suggested a TS model for NIDS ("Network Intrusion Detection Systems") with a RepTree algorithm. The evaluation of the model was conducted on the UNSW-NB15 dataset, as well as other datasets. In the 1st stage, the classifier categorized network traffic into UDP, TCP, and other categories, representing different network traces. Subsequently, the RepTree algorithm was employed in the second stage to classify anomalies and make attack predictions. To decrease the feature space, the researchers utilized the Information Gain (IG) and Consistency (IGC) methods. For binary classification, the TS-RepTree model obtained an accuracy of 88.95% on the test dataset, using approximately 20 relevant features. Nevertheless, it is significant to note that one limitation of the research has been the consideration of only one metric for evaluating the performance of the framework.

Gao J et al.[22] suggested an Intrusion Detection System (IDS) that employed an incremental method combining the ELM ("Extreme Learning Machine") with the APCA ("Advanced Principal Component") algorithm. The APCA algorithm was utilized to dynamically choose the most relevant features needed by the ELM for optimal attack prediction. The evaluation of the suggested IDS framework was conducted with the UNSW-NB15 dataset. The primary metric considered for performance evaluation was the accuracy of the test data, along with the DR ("Detection Rate") and FAR. The experimental findings demonstrated that the IELM-APCA achieved a false alarm rate of 35.09%, a detection rate of 77.36%, and an accuracy of 70.51%.

Almogren AS et al. [23] the authors presented a specialized Intrusion Detection System (IDS) called EoT-IDS, designed for the EoT ("Edge-of-Things") environment, which extends the IoT paradigm. The EoT-IDS incorporated a feature selection module that utilized the correlation approach to identify the most relevant attributes for intrusion detection. The selected subset of features was then used as input for a DBN ("Deep Belief Network") classifier to make predictions. The efficiency of the suggested system was assessed using the UNSW-NB15 dataset, with accuracy on the test data as the primary performance metric. The authors experimented with various configurations of DBN networks and identified the optimal configuration. This configuration consisted of 64 hidden units in layer 1, 60 hidden units in layer 2, and a DBN architecture serial number of 28. The EoT-IDS using this configuration obtained an accuracy of 85.73 percent for binary classification, showcasing its effectiveness in detecting intrusions in the EoT environment.

Jiang K et al. [24] suggested a framework for NIDS aimed at improving accuracy. The framework incorporated several techniques to improve the system's performance. To address noisy data records in the majority classes, the O-SS ("One-Side Selection") method

was applied, reducing the impact of such noise. Additionally, the SMOS ("Synthetic Minority Over Sampling") technique was employed to raise the number of minority cases in the dataset, thereby improving representation. For attribute extraction, spatial attributes were obtained using Convolutional Neural Networks (CNN), which excel at extracting features from data with spatial structure. Temporal attributes, on the other hand, were selected using Bi-LSTM ("Bidirectional Long-Short Term Memory") models, which are effective in capturing temporal dependencies in sequential data. The combination of Bi-LSTM and CNN served as the DL model for predictive tasks within the proposed framework. By leveraging the strengths of both architectures, the framework aimed to improve the accuracy of intrusion detection. To assess the performance of the framework, tests have been conducted with UNSW-NB15 & NSL-KDD intrusion detection datasets. The test data's accuracy served as the primary performance metric. The experimental findings showcased that the CNN-Bi-LSTM model achieved accuracies of 83.58% and 77.16% for the UNSW-NB15 as well as NSL-KDD datasets, respectively. These findings demonstrate the effectiveness of the framework in enhancing accuracy for intrusion detection tasks in network environments.

Sarumi, O.A et al. [25] conducted a comparative analysis of two different approaches, Apriori (a rule-based algorithm) and SVM ("Support Vector Machine") (a ML technique). The evaluation was performed on the datasets of UNSW-NB15 and NSL-KDD, and both filter-based and wrapper-based feature selection techniques were employed. The test findings on the dataset of NSL-KDD revealed the performance of the various approaches. The filter-SVM approach obtained a precision of 66.34%, a recall of 95.38%, and an accuracy of 77.17%. In comparison, the filter-Apriori method obtained a precision of 85.77%, a recall of 57.89%, and an accuracy of 67.00%. Moving to the wrapper-based techniques, the wrapper-SVM approach achieved a precision of 68.41%, a recall of 98.02%, and an accuracy of 79.65%. On the other hand, the wrapper-Apriori method obtained a precision of 85.79%, a recall of 58.81%, and an accuracy of 68%. These findings highlight the varying performance of the different approaches and emphasize the impact of feature selection techniques on the accuracy, recall, and precision of Intrusion Detection System (IDS) models. The findings suggest that the wrapper-SVM approach outperformed the other approaches in terms of accuracy, recall, and precision, indicating its effectiveness for intrusion detection tasks.

Almasoudy et al.[26] proposed a wrapper-based attribute selection technique for intrusion detection utilizing the differential evolution (DE) algorithm. The main objective was to select optimal feature sets for the extreme learning machine (ELM) classifier. The assessment of the suggested approach was conducted on the NSL-KDD dataset, which served as the experimental platform for DE-ELM. The study investigated both binary and multiclass classification scenarios. The experimental findings showcased the efficiency of the DE-ELM method. In the binary classification setup, the method acquired an accuracy of 80.15%. For the multiclass classification setup, the accuracy improved to 87.53%. These results underscore the potential of the suggested method in accurately identifying and classifying intrusions in network traffic. The authors also discussed future research directions, which include deploying the DE-ELM approach on real-time network traffic and enhancing its performance in dealing with underrepresented classes, with a specific focus on U2R attacks. By addressing these challenges, the DE-ELM approach can be further refined and applied to real-world intrusion detection systems, potentially improving their effectiveness in detecting and mitigating various types of intrusions.

Bostani, H et al.[27] the authors suggested a hybrid attribute selection method called BGSA-MI, which combines the binary gravitational search algorithm (BGSA) with the MI ("Mutual Information") approach. The aim was to enhance the performance of the standard BGSA by integrating it with the MI method as a filter algorithm. The effectiveness of the BGSA-MI approach has been assessed on the dataset of NSL-KDD intrusion detection. The fitness function of BGSA-MI was based on the SVM ("Support Vector Machine") classifier, with accuracy as the primary performance metric. For comparison, the Chi-square and ReliefF approaches were used as baseline algorithms. The test findings demonstrated that the BGSA-MI approach achieved an impressive accuracy score of 88.36%. In contrast, the ReliefF method achieved an accuracy of 84.60%, while the Chi-square method obtained an accuracy of 84.68%. These results highlight the superiority of the BGSA-MI approach in selecting the most relevant attributes for intrusion detection. The approach outperformed the ReliefF and Chi-square methods in terms of accuracy, indicating its effectiveness in improving the performance of the attribute selection process. The integration of BGSA and MI provides a powerful hybrid method for feature selection in IDSs.

Hosseini, S et al. [28] proposed a two-step intrusion detection system (IDS) that combines machine learning (ML) techniques. The first step involved feature selection using the logistic regression (LR) algorithm and the genetic algorithm (GA). The goal was to detect the most

significant features for intrusion detection. In the 2nd phase, the authors employed the ANN ("Artificial Neural Network") for classification, leveraging evolutionary-based algorithms such as PSO to enhance the training process of the ANN. The performance of the suggested frameworks has been assessed on the NSL-KDD dataset, focusing on binary classification. The primary metrics considered were accuracy and training time. The experimental findings indicated that the PSO-ANN framework obtained an accuracy of 88.90% and completed the training process in 74 seconds. On the other hand, the GA-ANN framework achieved an accuracy of 83.11% with a training time of 134 seconds. These findings highlight the effectiveness of the proposed two-step IDS system. The combination of LR and GA for feature selection, along with the utilization of ANN with evolutionary-based algorithms for classification, yielded promising results. The PSO-ANN framework demonstrated superior performance in terms of greater accuracy and faster training time compared to the GA-ANN framework. This study contributes to the development of efficient IDS systems by integrating feature selection and classification techniques from the field of ML.

Zhang, C et al. [29] proposed a DL-based IDS with NSL-KDD dataset. Their approach involved the utilization of an AE ("Autoencoder") for attribute extraction and a deep neural network (DNN) for classification. The evaluation of the AE-DNN approach considered multiple performance metrics, including F1-score, recall, precision, and accuracy. These metrics offer insights into the ability of the system to correctly classify intrusions and detect true positives while minimizing false positives and false negatives. The experimental findings showed that the AE-DNN approach achieved a classification accuracy of 79.74%. This indicates that approximately 79.74% of the instances were correctly classified by the IDS. The precision value of 82.22% suggests that a high percentage of the instances classified as intrusions were indeed true positives, reducing the number of false positives. Moreover, the recall value of 79.74% indicates that the IDS was able to identify a significant portion of the actual intrusions in the dataset. The F1-score of 76.47% represents a balance between precision and recall, combining both metrics to provide an overall assessment of the IDS's performance. These findings highlight the efficiency of the DL-based IDS approach using the AE-DNN architecture. The system demonstrates the potential to accurately classify intrusions in the NSL-KDD dataset, with competitive performance in terms of "recall, precision, F1-score, and accuracy".

Wang, L et al. [30], the authors proposed a novel approach for intrusion detection systems (IDS) that aimed to address the class imbalance and mitigate the effect of irrelevant information during training. Their framework, named AS-CNN, combined the adaptive synthetic sampling (ADASYN) approach with a CNN.The ADASYN technique was employed to handle class imbalance by generating synthetic samples for the minority class, thus balancing the representation of different classes in the dataset. The CNN algorithm was enhanced with a split convolution module (SPCCNN) to better capture relevant features for intrusion detection. To assess the AS-CNN framework performance, the authors conducted experiments using the NSL-KDD dataset. They compared their method against two baseline models: the RNN ("Recurrent Neural Network") and the simple CNN. The primary performance metric considered in their evaluation was detection accuracy. The experimental findings indicated that the RNN obtained an identification accuracy of 69.73%, while the CNN obtained an accuracy score of 68.66%. In contrast, the AS-CNN approach obtained a notable accuracy of 80%. These findings demonstrate the effectiveness of the AS-CNN framework in improving the accuracy of intrusion detection compared to the baseline models. By integrating the ADASYN technique to handle class imbalance and the SPCCNN module to capture relevant features, the AS-CNN framework shows promise in enhancing the performance of intrusion detection systems. The significant improvement in accuracy achieved by AS-CNN suggests its potential for more accurate and reliable detection of intrusions in real-world scenarios.

3. Unsw-Nb15 Dataset:

For our experimental procedures, we utilized the UNSW-NB15 attacks dataset [39]. This dataset originally consisted of 45 features, as detailed in Table 1. Among these features, 4 instances were non-numeric (categorical), while 41 were numeric. As part of our research, we further partitioned the UNSW-NB15 subset into 2 parts: UNSW-NB15-TRAIN, which comprised 70% of the original training set obtained from the dataset of UNSW-NB15, and UNSW-NB15-TEST, which consisted of the remaining 30% of the original testing set from the dataset of UNSW-NB15.

Table 1: List features of the UNSW-NB15 Dataset

Sno	Feature	Dtype	Sno	Feature	Dtype
1	id	int64	24	dwin	int64
2	dur	float64	25	tcprtt	float64
3	proto	object	26	synack	float64

4	service	object	27	ackdat	float64
5	state	object	28	smean	int64
6	spkts	int64	29	dmean	int64
7	dpkts	int64	30	trans_depth	int64
8	sbytes	int64	31	response_body_len	int64
9	dbytes	int64	32	ct_srv_src	int64
10	rate	float64	33	ct_state_ttl	int64
11	sttl	int64	34	ct_dst_ltm	int64
12	dttl	int64	35	ct_src_dport_ltm	int64
13	sload	float64	36	ct_dst_sport_ltm	int64
14	dload	float64	37	ct_dst_src_ltm	int64
15	sloss	int64	38	is_ftp_login	int64
16	dloss	int64	39	ct_ftp_cmd	int64
17	sinpkt	float64	40	ct_flw_http_mthd	int64
18	dinpkt	float64	41	ct_src_ltm	int64
19	sjit	float64	42	ct_srv_dst	int64
20	djit	float64	43	is_sm_ips_ports	int64
21	swin	int64	44	attack_cat	object
22	stcpb	int64	45	label	int64
23	dtcpb	int64			

The dataset of UNSW-NB15 comprises instances that are categorized into different forms of network attacks, including Fuzzers, Worms, Generic, DoS, Backdoor, Reconnaissance, Shellcode, Analysis, and Exploits. Table 2 in the dataset provides a detailed overview of each attack class, presenting comprehensive information and the distribution of values within various data subsets.

Table 2: Types of attacks in the UNSW-NB15 Dataset

Attack Type	UNSW- NB15	UNSW-NB15-TRAIN-1	UNSW-NB15-VAL	UNSW-NB15-TEST
Normal	56,000	41,911	14,089	37,000
Generic	40,000	30,081	9919	18,871
Exploits	33,393	25,034	8359	11,132
Fuzzers	18,184	13,608	4576	6062
DoS	12,264	9237	3027	4089
Reconnaissance	10,491	7875	2616	3496
Analysis	2000	1477	523	677
Backdoor	1746	1330	416	583
Shellcode	1133	854	279	378
Worms	130	99	31	44

4. Background on Ensemble Machine Learning Algorithms

In this study, we employed a set of ensemble methods called tree-based classifiers. The specific algorithms utilized in our research included RF, XGBoost, ET ("Extra-Trees"), and Adaptive Boosting (AdaBoost).

4.1. Random Forest

RF is a supervised ML algorithm that leverages the power of multiple DTs to make accurate predictions. In the RF approach, every DT independently generates its prediction, and the final prediction of the RF model is examined by the majority vote among the DTs. To train the RF model, a technique called bootstrapping is used, which involves creating unique subsets of the training data for each DT in the RF. Additionally, RF employs a greedy algorithm to select optimal split points during the construction of the decision trees [31, 32].

4.2. Extra-Trees

ET also-referred to as "Extremely Randomized Trees", is another supervised machine learning method that utilizes multiple Decision Trees (DTs) for decision-making. ET can be applied to both classification & regression problems. Unlike RF, where every DT is built from a subset of the training set, ET fits individual DTs using the entire training set. Additionally, the ET approach randomly selects split points for each node, further enhancing the randomness of the tree construction process [33, 34].

4.3 XGBoost

The Extreme Gradient Boosting (XGBoost) algorithm is a powerful boosting technique that improves the performance of tree-based algorithms. While XGBoost,

Extra Trees (ET), and Random Forest (RF) are all ensemble methods in machine learning, there are significant distinctions between XGBoost and RF/ET. In XGBoost, decision trees are not independent entities; instead, each decision tree is built by extending an existing tree, learning from the mistakes made by the previous trees in the ensemble. This sequential learning process allows XGBoost to effectively capture complex patterns and improve predictive accuracy [35, 36].

4.4 AdaBoost

Adaptive Boosting, or AdaBoost, is a ML technique that iteratively develops a series of weak classifiers using various weighted subsets of the training data. Each weak classifier is a straightforward model, including a decision stump, that outperforms random guessing by a small margin (a decision tree with a single split). Based on how well the weak classifier performs, AdaBoost modifies the weights given to the training samples throughout each iteration. The weights are increased for the misclassified examples, making them more important in the subsequent iterations, and decreased for the correctly classified examples. By combining the predictions of these weak classifiers using a weighted majority vote, AdaBoost creates a strong ensemble classifier that can accurately classify instances in the dataset [37]

4.5 Stacking

Stacking, introduced by David H. Wolpert [38], is an ensemble strategy designed to reduce error rates in generalization. Its objective is achieved by combining multiple primary learners and incorporating meta-learners into the ensemble. The following steps outline the process of stacking (refer to Figure 1). The dataset is initially divided into a training dataset and a testing dataset. The training dataset is then divided into K equal parts, typically using a technique called K-fold cross-validation. For example, if K is set to five, the dataset is split into five parts of equal size. In the stacking approach, four parts of the training dataset are used to train the primary learner, which can be any machine learning algorithm. The primary learner learns to make predictions based on the input features and target labels in these four parts. The remaining part of the training dataset, also known as the validation set, is then used to generate predictions using the trained primary learner. These predicted values serve as the new input features, along with the original input features, for the meta-learner. The meta learner, which can be another machine learning algorithm, is trained using the predicted values from the primary learner as input features and the actual target labels from the validation set. It learns to make predictions based on this combined information. Once the stacking process is trained, it can be applied to predict and classify new instances or data points. The primary learner is used to generate predictions on the new data, and these predictions are then fed into the meta learner, which produces the final prediction or risk estimation for the attacks. The stacking approach allows for combining the strengths of multiple models by using one model's predictions as input for another model. This can often result in improved performance and more accurate predictions compared to using a single model alone.

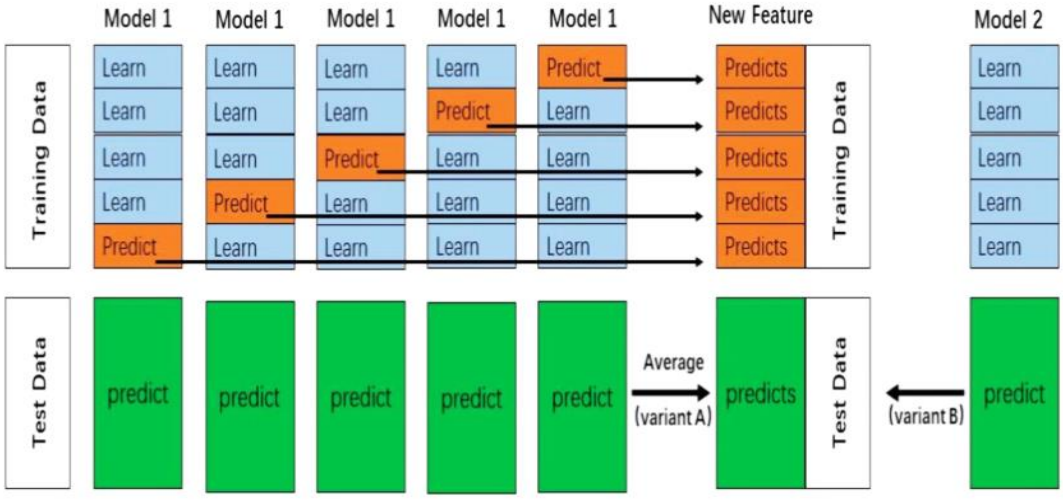

Fig. 1: The process of stacking

5. Proposed Methodology

In this section, we express a detailed explanation of the IDS model called GA-EM's. The complete workflow of the model is illustrated in Figure 2. Firstly, pre-processing operations are performed after loading the UNSW-NB15 dataset. To enhance accuracy, a feature selection process is conducted using the GA algorithm, which identifies the most relevant features. Once the optimal training dataset, as well as the feature subset, are determined, they are used in the classifier training phase, where the EM's algorithm is employed to classify instances as either anomalous or

normal. Finally, the GA-EM's model evaluates its performance based on accuracy alone.

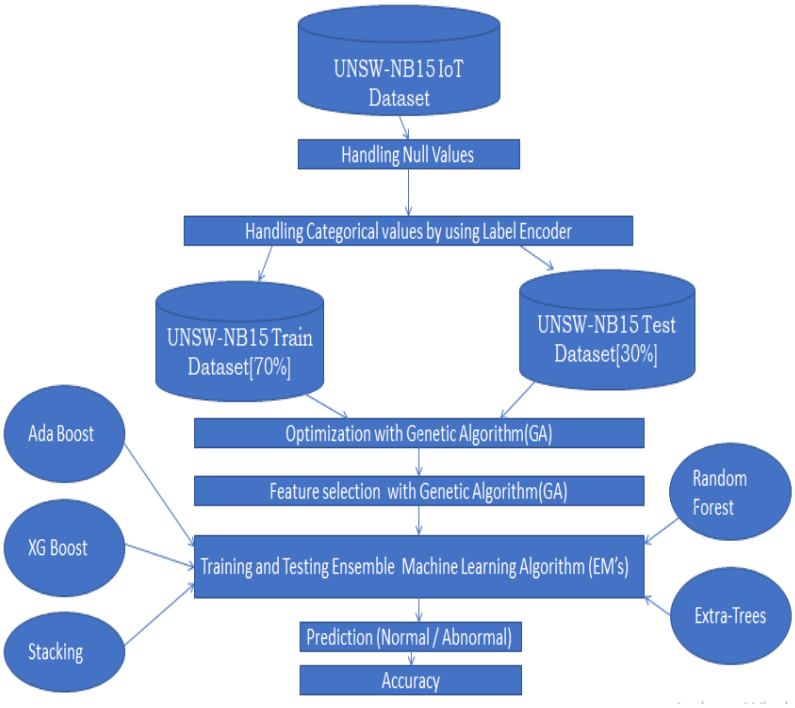

Fig. 2: Flow of the proposed GA-EM's Model

5.1 Feature selection using GA (Genetic Algorithm)

GA is a search and optimization technique inspired by the natural selection and genetics process. It is used to develop optimum or near-optimal solutions to complicated problems by mimicking the observable evolutionary principles in nature. The algorithm starts with a population of potential solutions, represented as individuals or chromosomes. Each chromosome is composed of genes or variables that encode a potential solution. The fitness function evaluates how well each individual performs in solving the problem, assigning a fitness score to measure its quality.

The GA operates through a series of iterative steps called generations. In each generation, the algorithm applies genetic operators to create new offspring individuals from the existing population. These operators include selection, crossover, and mutation.

Selection: Higher fitness scores individuals are more likely to be chosen as parents for reproduction. This mimics the principle of "survival of the fittest," where individuals with better solutions have a greater probability of passing their traits to the next generation.

Crossover: Crossover is the genetic operator that combines genetic material from two parent chromosomes to create offspring. It promotes the exchange of genetic information and can lead to the creation of new and potentially better solutions. The genetic operator known as mutation also delivers random alterations to certain chromosomes.

Mutation: Mutation introduces random modifications in the genetic information of individuals. It supports introducing diversity into the population and avoids premature convergence to suboptimal solutions. By randomly modifying a gene or set of genes in an individual, the algorithm explores new regions of the search space. After creating the offspring individuals through crossover and mutation, they form the next generation. This process continues for a fixed number of generations or until a termination criterion is met (e.g., reaching a satisfactory solution or exceeding a predetermined number of iterations). Over successive generations, the population tends to evolve and converge toward better solutions. The fittest individuals in the final generation are considered the optimal/ near-optimal solutions to the problem. It helps maintain diversity in the population and prevents premature convergence to suboptimal solutions. Through successive iterations of selection, crossover, and mutation, the GA converges towards better solutions over time. The process keeps on until a stopping requirement is satisfied, such as when the required number of generations or level of solution quality is reached. The overall process of GA in figure-3.

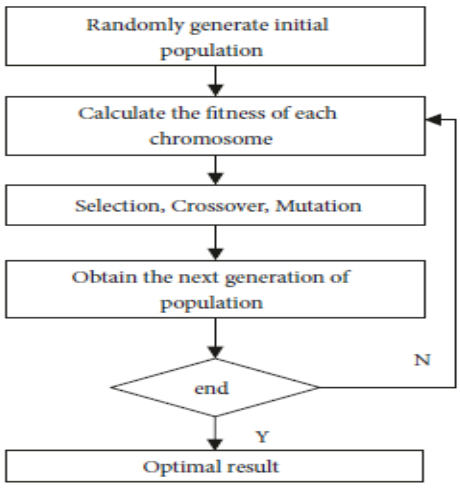

Fig. 3: Process of GA.

Here's a Pseudo code representation of a basic Genetic Algorithm (GA):

```
Genetic algorithm parameters:
    Population size
    Mating pool size
    Number of mutations

sol_per_pop = 30  // Population size
num_parents_mating = 20  // Number of parents inside the mating pool
num_mutations = 20  // Number of elements to mutate

// Defining the population shape
pop_shape = (sol_per_pop, num_features)  // Population shape based on the number of features

// Creating the initial population
population = randomly_initialize_population(pop_shape)

best_outputs = []

num_generations = 50  // Number of generations

for generation in range(num_generations):
    // Evaluate the fitness of each individual in the population
    fitness = evaluate_fitness(population, X_train, y_train, X_test, y_test)

    // Store the best fitness value of this generation
    best_outputs.append(np.max(fitness))

    // Select parents for mating
    parents = select_mating_pool(population, fitness, num_parents_mating)

    // Create offspring through crossover
    offspring = crossover(parents, pop_shape)

    // Apply mutation to the offspring
    offspring_mutated = mutation(offspring, num_mutations)

    // Replace the old population with the new population (offspring)
    population = offspring_mutated

// Find the best solution in the final population
best_solution_idx = np.argmax(fitness)
best_solution = population[best_solution_idx]

print("Best solution:", best_solution)
print("Best solution fitness:", fitness[best_solution_idx])
```

Algorithm 1: Process of Pseudo code of the standard basic algorithm (GA).

After applying the Genetic Algorithm (GA), the original dataset with 43 features underwent feature selection. The GA process effectively reduced the number of features to 21, resulting in a more optimized and concise feature set for further analysis.

5.2. Classifier Training

We can obtain the optimal training dataset as well as the optimal subset of features based on data sampling and feature selection. The optimal training set undergoes dimension reduction based on the selected feature subset. As the Ensemble machine learning algorithms (EM's) classifier is capable of handling binary classification problems, we can utilize it to identify one class of attack behavior and another one is normal behavior class for every class, data sampling & feature selection approaches are applied to obtain the optimal training dataset as well as feature subset specific to that class.

6. Results And Discussion:

The intrusion detection algorithm GA-EM has shown remarkable performance on the UNSW-NB15 dataset, which was sourced from "https://research.unsw.edu.au/projects/unsw-nb15-dataset[40]". This dataset was meticulously created by the UNSW Canberra Cyber Range Lab, capturing raw network packets to simulate modern attack behaviors and generate real-world normal activities. The UNSW-NB15 dataset holds immense value as a valuable resource for conducting cybersecurity research and analysis.

6.1 Confusion matrix & Accuracy

You have provided a clear explanation of a confusion matrix and accuracy in the context of a classification model and intrusion detection system. A confusion matrix shown in Table 3 assists assess the performance of a model by showing the "true positive, true negative, false positive, as well as false negative" predictions. It provides a breakdown of the model's performance for each class (in this case, "Abnormal" and "Normal").

Table 3: Confusion matrix

	Predicated Negative	Predicated positive
Abnormal	TN	FP
Normal	FN	TP

Accuracy, as you described, measures the total correctness of the model's predictions. It is determined by dividing the sum of true positive & negative predictions by the sum of true negative, true positive, false positive, as well as false negative predictions. The model's ability to accurately categorize both attacks and normal instances is shown by accuracy.

"Accuracy=(True Positive+True Negative)/(True Positive+True Negative+False Positive+False Negative)"

This formula presents the proportion of correctly classified instances (both attacks and non-attacks) to the total number of cases in the dataset. It provides an overall measure of the IDS's ability to accurately identify and classify network traffic, taking into account both correct classifications of attacks and non-attacks. A higher accuracy value indicates a more reliable and effective intrusion detection system.

The results of the proposed system's GA-EM's are provided in Table-4. Various types of Ensemble machine-learning techniques have yielded different accuracy rates.

Table 4: Accuracy of Ensemble machine learning algorithms (EM's)

EM's method	Accuracy rates
Random forests	98.06%
XGBoost	97.99%
Extra-Trees	97.80%
AdaBoost	97.57%
stacking	97.30%

These accuracy rates represent the performance of different machine learning algorithms (Random Forests, XGBoost, AdaBoost, Extra-Trees, and Stacking) when applied to EM's method. Each algorithm achieved varying levels of accuracy in predicting the target variable. Random Forests exhibited the highest accuracy rate of 98.06%, followed by XGBoost with 97.99% accuracy. Extra-Trees achieved an accuracy of 97.80%, while Adaboost and stacking achieved accuracies of 97.57% and 97.30%, respectively. The overall Accuracy of GA-EM's ,bar plot for Analysed performance in Figure 4.

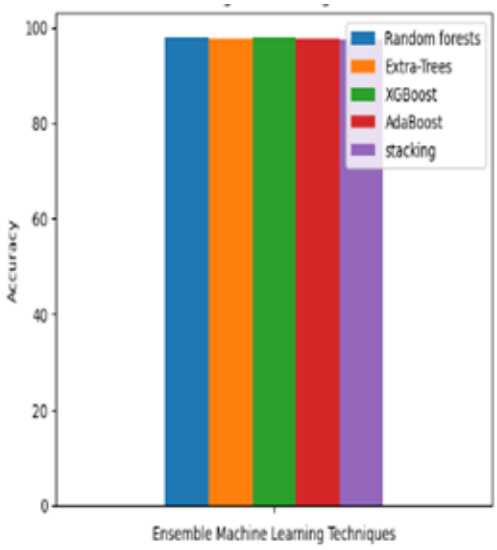

Fig. 4: Bar plot for Analysed performance

7. Conclusion

In this work, we have suggested an intrusion detection model that combines GA for feature selection and Ensemble machine learning algorithms (EM's) for classification. The finding of our work shows the efficiency of this model in enhancing the accuracy and detection rate of intrusion detection while minimizing the number of features required. The GA-based feature selection process helps identify an optimal subset of features, leading to improved accuracy and more efficient detection of intrusions. In the GA-EM's framework, different Ensemble Machine Learning techniques were evaluated for their performance. Random forest emerged as the top-performing technique, achieving the highest accuracy rate of 98.06%. Close behind is XGboost with an accuracy rate of 97.99%. Extra-Trees and AdaBoost also exhibited competitive accuracy rates of 97.80% and 97.57%, respectively. Stacking achieved an accuracy rate of 97.30%. These results demonstrate the varying performance levels of the different ensemble techniques within the GA-EM's framework. Overall, our proposed GA-EM model shows promise in enhancing the intrusion detection process by effectively selecting features and employing powerful ensemble techniques. The results highlight the importance of choosing the right combination of techniques to achieve optimal accuracy rates and efficient detection of intrusions.

In future studies, we plan to incorporate a synthetic oversampling method to address the imbalance issue in the UNSW-NB15 IoT dataset. This method aims to raise the presentation of minority classes in the training process, thereby enhancing the performance of the model in identifying these classes. Additionally, we intend to explore the application of multiclassification techniques to handle different forms of attacks present within the UNSW-NB15 IoT dataset. By employing these techniques, we aim to enhance the overall accuracy and robustness of the intrusion detection system in a wider range of attack scenarios.

References

[1] K. Lueth, "State of the IoT 2018: Number of IoT devices now at 7B – Market accelerating." https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b /(accessed May 27, 2020).

[2] R. McKay, B. Pendleton, and J. Britt, "Machine Learning Algorithms on Botnet Traffic: Ensemble and Simple Algorithms," Proceedings of the 2019 3rd International Conference on Compute and Data Analysis, p. 5, 2019.

[3] M. Aldwairi, W. Mardini, A. Alhowaide, Anomaly Payload Signature Generation System Based on Efficient Tokenization Methodology, International Journal on Communications Antenna and Propagation (IRECAP) (2018) (Nov. 2018).

[4] T. Mohamed, T. Otsuka, T. Ito, Towards Machine Learning Based IoT Intrusion Detection Service," Recent Trends and Future Technology in Applied Intelligence. IEA/AIE 2018, Lecture Notes in Computer Science 10868 (May 2018), https://doi.org/10.1007/978-3-319-92058-0_56.

[5] I. Butun, S.D. Morgera, R. Sankar, A Survey of Intrusion Detection Systems in Wireless Sensor

Networks, IEEE Communications Surveys Tutorials 16 (1) (2014) 266–282, https://doi.org/10.1109/SURV.2013.050113.00191. First.

[6] C. Zhang, Y. Ma (Eds.), Ensemble Machine Learning: Methods and Applications, Springer-Verlag, New York, 2012, https://doi.org/10.1007/978-1-4419-9326- 7.

[7] S. Raschka, Python Machine Learning - Second Edition, Packt Publishing, 2017. Accessed: Nov. 19, 2020.

[8] . Khammassi C, Krichen S. A GA-LR wrapper approach for feature selection in network intrusion detection. Comput Secur 2017;70:255–77.

[9] Osanaiye O, Cai H, Choo K-KR, Dehghantanha A, Xu Z, Dlodlo M. Ensemble-based multi-flter feature selection method for DDOS detection in cloud computing. EURASIP J Wirel Commun Netw. 2016;20]16(1):130.

[10] Ambusaidi MA, He X, Nanda P, Tan Z. Building an intrusion detection system using a flter-based feature selection algorithm. IEEE Trans Comput. 2016; 65(10):2986–98.

[11] Y. Zhou, M. Han, L. Liu, J.S. He, Y. Wang, Deep learning approach for cyberattack detection, in: IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Apr. 2018, pp. 262–267, https://doi.org/10.1109/INFCOMW.2018.8407032.

[12] S.T. Miller, C. Busby-Earle, Multi-Perspective Machine Learning a Classifier Ensemble Method for Intrusion Detection, in: Proceedings of the 2017 International Conference on Machine Learning and Soft Computing - ICMLSC '17, Ho Chi Minh City, Vietnam, 2017, pp. 7–12, https://doi.org/10.1145/3036290.3036303.

[13] B.A. Tama, M. Comuzzi, K.-H. Rhee, TSE-IDS: A Two-Stage Classifier Ensemble for Intelligent Anomaly-Based Intrusion Detection System, IEEE Access 7 (Jul. 2019) 94497–94507, https://doi.org/10.1109/ACCESS.2019.2928048.

[14] M. Aloqaily, S. Otosum, I.A. Ridhawi, Y. Jararweh, An intrusion detection system for connected vehicles in smart cities, Ad Hoc Networks 90 (Jul. 2019), 101842, https://doi.org/10.1016/j.adhoc.2019.02.001.

[15] A.J. Siddiqui, A. Boukerche, TempoCode-IoT: temporal codebook-based encoding of flow features for intrusion detection in Internet of Things, Cluster Comput (Sep. 2020), https://doi.org/10.1007/s10586-020-03153-8.

[16] Connelly L. Logistic regression. Medsurg Nurs. 2020;29(5):353–4.

[17] Gao J, Chai S, Zhang B, Xia Y. Research on network intrusion detection based on incremental extreme learning machine and adaptive principal component analysis. Energies 2019;12(7):1223.

[18] Almogren AS. Intrusion detection in edge-of-things computing. J Parallel Distrib Comput. 2020;137:259–65.

[19] Jiang K, Wang W, Wang A, Wu H. Network intrusion detection combined hybrid sampling with deep hierarchical network. IEEE Access. 2020; 8:32464–476

[20] [20]. Khan NM, Negi A, Thaseen IS, et al. Analysis on improving the performance of machine learning models using feature selection technique. In: International conference on intelligent systems design and applications. Springer; 2018. pp. 69–77

[21] Huibing Wang, Jinbo Xiong, Zhiqiang Yao, Mingwei Lin, and Jun Ren. Research survey on support vector machine. In Proceedings of the 10th EAI International Conference on Mobile Multimedia Communications, pages 95–103, 2017.

[22] Mohammad Marufur Rahman, Md Islam, Md Manik, Motaleb Hossen, Mabrook S Al-Rakhami, et al. Machine learning approaches for tackling novel coronavirus (covid-19) pandemic. Sn Computer Science, 2(5):1–10, 2021.

[23] Mr Brijain, R Patel, Mr Kushik, and K Rana. A survey on decision tree algorithm for classifcation. International Journal of Engineering Development and Research, IJEDR, 2(1), 2014.

[24] Breiman L. Random forests Machine learning. 2001;45(1):5–32.

[25] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Random forests. In The elements of statistical learning, pages 587–604. Springer, 2009.

[26] Belouch M, El Hadaj S, Idhammad M. A two-stage classifer approach using reptree algorithm for network intrusion detection. Int J Adv Comput Sci Appl. 2017;8(6):389–94

[27] Gao J, Chai S, Zhang B, Xia Y. Research on network intrusion detection based on incremental extreme learning machine and adaptive principal component analysis. Energies 2019;12(7):1223.

[28] Ahmad, M.W.; Reynolds, J. and Rezgui; Y. Predictive modelling for solar thermal energy systems: A comparison of support vector regression, random forest, extra trees and regression trees. Journal of cleaner production, 2 ¯ 018, 203, 810–821.

[29] Alsariera, Y.A.; Adeyemo, V.E.; Balogun, A.O. and Alazzawi, A.K. AI meta-learners and extra-trees algorithm for the detection of phishing websites. IEEE Access, 2 ¯ 020, 8, 142532–142542.

[30] Devan, P. and Khare, N., 2020. An efficient XGBoost–DNN-based classification model for network intrusion detection system. Neural Computing and Applications,2 ¯ 020, 1–16.

[31] Scikit-Learn: Ensemble Gradient Boosting Classifier. Available online: https://scikit-learn.org/stable/modules/generated/ sklearn.ensemble.GradientBoostingClassifier.html (accessed on 21 May 2021

[32] Wolpert, D.H. Stacked generalization. Neural Netw. **1992**, 5, 241–259

[33] F. Amato, N. Mazzocca, F. Moscato and E. Vivenzio, "Multilayer Perceptron: An Intelligent Model for Classification and Intrusion Detection," 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA), Taipei, Taiwan, 2017, pp. 686-691, doi: 10.1109/WAINA.2017.134.

[34] Moustafa N, Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 military communications and information systems conference (MilCIS). IEEE; 2015. pp. 1–6.

[35] Anwer, H. M., Farouk, M., & Abdel-Hamid, A. (2018, April). A framework for efficient network anomaly intrusion detection with features selection. In 2018 9th International Conference on Information and Communication Systems (ICICS) (pp. 157-162). IEEE

[36] Hauke, J., & Kossowski, T, Correlations between variables can be measured with the use of different indices (coefficients). The three most popular are: Pearson's coefficient, Spearman's rho coefficient, and Kendall's tau coefficient (2011)

[37] Scikit Learn, Machine Learning in Python. https://scikit-learn.org/stable. Accessed 26 Sept 2020.

[38] Kapoor, E. ., Kumar, A. ., & Singh , D. . (2023). Energy-Efficient Flexible Flow Shop Scheduling With Due Date and Total Flow Time. International Journal on Recent and Innovation Trends in Computing and Communication, 11(2s), 259–267. https://doi.org/10.17762/ijritcc.v11i2s.6145

[39] Omondi, P., Rosenberg, D., Almeida, G., Soo-min, K., & Kato, Y. A Comparative Analysis of Deep Learning Models for Image Classification. Kuwait Journal of Machine Learning, 1(3). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/128

[40] Soundararajan, R., Stanislaus, P. M., Ramasamy, S. G., Dhabliya, D., Deshpande, V., Sehar, S., & Bavirisetti, D. P. (2023). Multi-channel assessment policies for energy-efficient data transmission in wireless underground sensor networks. Energies, 16(5) doi:10.3390/en16052285 Talukdar, V., Dhabliya, D., Kumar, B., Talukdar, S. B., Ahamad, S., & Gupta, A. (2022).