Journal of Applied Artificial Intelligence
https://jaai.sabapub.com
ISSN: 2709-5908

2022 Volume 3, Issue 1 : 1 – 16
DOI : 10.48185/jaai.v3i1.450

E2IDS: An Enhanced Intelligent Intrusion Detection System Based On Decision Tree Algorithm

Mohamed Aly Bouke[1,*], Azizol Abdullah[1], Sameer Hamoud ALshatebi[1], Mohd Taufik Abdullah[1]

[1]Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang 43400.

Received: 04.03.2022 • Accepted: 30.04.2022 • Published: 30.06.2022 • Final Version: 30.06.2022

Abstract: Due to the increased usage of the Internet of Things and heterogeneous distributed devices, the development of effective and reliable intrusion detection systems (IDS) has become more critical. The massive volume of data with various dimensions and security features, on the other hand, can influence detection accuracy and raise the computation complexity of these systems. Fortunately, Artificial Intelligence (AI) has recently attracted a lot of attention, and it is now a principal component of these systems. This work presents an enhanced intelligent intrusion detection model (E2IDS) to detect state of the art known cyberattacks. The model design is Decision Tree (DT) algorithm-based, with an approach to data balancing since the data set used is highly unbalanced and one more approach for feature selection. Furthermore, accuracy, recall and F-score are selected as the performance evaluation metrics. The experimental results show that our E2IDS not only overcomes the benchmark work but also reduces the complexity of the computing process.

Keywords: Security features Importance, Decision Trees (DT), Machine learning (ML), and Anomaly Based IDS

1. Introduction

Authors In our world today, the Internet has affected humanity in a significant manner, as it has become more abyssal in daily life. However, it is also exposing us to progressively severe security threats. Given the widespread adoption of network access, the market of network security is on the rise. In today's network protection, IDS is an essential component. Due to the rapid growth of the attack surface, protection methods currently in use, such as Firewalls, user authentication, and data encryption, aren't enough to keep networks safe. IDS can monitor and analyze network traffic, detect security threats, and preserve the data's confidentiality, integrity, and availability [1]. IDS can be divided into two tiers according to their functional approach: Knowledge-based and Behaviour-based. The Knowledge-based detection category seeks to identify attacks by examining their signatures; They're used to detect known forms of attacks while keeping a low false alarms rate [2]. However, this approach requires extra effort to update database rules and signatures for detection performance. Since zero-day attack does not have any signature, this technique cannot detect them. Secondly, The Behaviour-based detection category assumes that the intruder's behavior differs from that of the rest of the network. [3]. In detail, anomaly-based detection builds a profile representing regular network traffic. It then identifies network activity deviations from this profile to detect attacks. Contrarily to the knowledge-based, this technique can detect zero-day attacks.

* Corresponding Author: muhammadalybuka@gmail.com

AI and ML have become essential technologies in modern information security architectures. Because of their ability to quickly analyze massive network events and identify a wide range of threats. Over time, these technologies are adapted more and more to build earlier incident detection systems that can handle any new variant of network attacks. User, asset, and network profiles are created using behaviour histories to allow AI to detect and respond to abnormal activities. One of the ML algorithms used to detect malicious activities is DT. DT can examine an extensive collection of intrusion detection data and find the key network features that indicate malicious activity. It can also see tendencies that might aid further research, extraction of attack features, and network monitoring duties[4].

The key benefit of using DT over other classification algorithms is due to its comprehensive logical collection of simple rules, and it can be easily integrated with real-time technologies [5]. The DT algorithm is a supervised learning (SL) algorithms group member. However, unlike other SL algorithms, the DT approach can address multi-classification problems(Classification and Regression). There are several versions of DTs, such as Iterative Dichotomiser 3 (ID3), and C4.5. ID3 relies on Entropy and employs an information theory mainly for feature ranking. On the other hand, The C4.5 algorithm was created to support Classification and Regression Trees (CART) [6].

This paper proposes an enhanced ML-based intrusion Detection model that targets a broad range of intrusion attacks. Factors such as the importance of security features and data balancing are given high priority in this work. After balancing the dataset, The most important features are selected to build the intrusion detection model.

The rest of the paper is organized as follows. Section 2 looks into similar work on intrusion detection models. Section 3 introduces and describes our E2IDS model. Experiments and evaluations of the suggested model are presented in Section 4. The article's final section brings the study to a close and emphasizes our future efforts.

2. Literature Review

AI can help security teams keep ahead of the threats as cyberattacks become more intense and frequent. Furthermore, it predicts security threats by processing vast amounts of data, abbreviating critical decision-making time, and responding to security incidences. IDS is usually used to reveal harmful cyber-attack activity on a network by tracking and reviewing regular network or computer system operations [7]. Plenty of research has been done in this context, involving various ML and Deep Learning (DL) techniques to build IDSs. DL-based methods have recently been used in IDSs and have shown to be efficient at detecting intrusive behaviours [8]–[11]. Ferrag et al. [12] presented a high-quality comparative study survey of DL methods for cybersecurity intrusion detection and the datasets used. They covered more than 30 popular datasets used to build IDSs. Zhao et al. [13] proposed an effective DL-based intelligent intrusion detection (IID) method using federated learning (FL) and aided long short-term memory (FL-LSTM) framework. The proposed model showed promising results compared to the conventional neural network (CNN).

Wang et al. [14] proposed an integrated deep intrusion model SDAE-ELM and DBN-SoftMax, based on the deep Denoising Autoencoder and Deep Belief Network(DBN). The model can handle real-time data, analyze large-scale data, and reduce the noise in the dataset. Zhao, Zhang, and Zheng [15] Proposed a DBN and probabilistic neural network (PNN) based intrusion detection model. The proposed approach uses DBN to reduce PNN network training and testing time by translating raw data into low-dimensional data. Spinelli et al. [16] suggested a behaviour-based Network Intrusion Detection System Near-to-real time (ANIDINR), a detailed methodology to predict zero-day attacks.

However, the proposed scheme generates an incorrect response when the attacker produces an attack profile close to the normal.

One of the most popular and broadly used techniques for developing IDS models is classification. Many standard classification algorithms have been proposed in recent decades, including Support Vector Machines (SVM), Naive Bayes (NB), K-Nearest Neighbor (KNN), Random Forest (RF), Logistic Regression (LR), and artificial neural network (ANN) according to Sarker and Kayes [17]. Likewise, feature selection (FS) is essential in creating ML models; many works have focused on this step. Amiri et al. [18] introduced a new IDS that uses knowledge theoretic and statistical principles for FS and the least square SVM (LSSVM) approach for classification. Experiments using the Kddcup99 dataset showed a significant boost in classification accuracy. Gu and Lu [19] used SVM with naïve Bayes feature embedding to design a practical intrusion detection model. Garg and Batra [20] Advised a hybrid behaviour-based detection method and the Fuzzified Cuckoo-based Clustering Technique (F-CBCT), which can identify the anomalies with high Detection Rates and reduced false-positive Rate. Kuang et al. [21] Designed a novel IDS model using the SVM and kernel principal component analysis (KPCA) combined with a genetic algorithm (GA). The proposed model uses KPCA to extract the main features.

Ambusaidi et al. [22] Proposed Flexible Mutual Information FS, a supervised filter-based FS algorithm. Li et al. [23] used the most common Kddcup99 dataset to build a hyperplane-based SVM classifier with a radial basis kernel function (N-RBF) to classify a few predefined attack categories. Kang and Kim [24] suggested an algorithm for selecting the best features. The proposed algorithm is based on a local search algorithm, a typical meta-heuristic algorithm for solving computationally tricky optimization problems. S. Mohammadi, Mirvaziri, Ghazizadeh-Ahsaee, et al. [25] suggested a novel filter-based FS algorithm to improve detection rates while lowering false-positive rates. Koc et al. [28] advised a naive Bayes-based classifier to construct a multiclass IDS. The KNN algorithm is another popular ML algorithm.

Sarker et al. recently proposed [26] a BehavDT ML method to create a user-centric context-aware prognostic framework using a behavioural DT and [27] an Intrusion Detection Tree ("IntruDTree") ML-based IDS model supported by a well-designed feature ranking and selection approach. Aloqaily et al. [28] proposed D2H-IDS, a new hybrid approach for intrusion detection in intelligent connected vehicle cloud environments. Eesa et al. [29] Proposed a novel combination method based on ID3 and the bees algorithm (BA) for subset FS. Puthran and Shah [30] proposed an ameliorated DT algorithm using binary split (IDTBS) and an enhanced DT algorithm using quad split (IDTQS) for increasing the detection performance for a predefined set of attacks such as Probe, U2R, and R2L using the KDD99 dataset. Al-Omari et al. [31] presented an intelligent tree-based model capable of predicting and detecting cyber-attacks efficiently and effectively. Ingre et al. [32] used the NSL-KDD to design DT-based IDS.

DT can analyze data and detect abnormal activities. The key benefit of adopting DT over other techniques is that they provide a comprehensive set of simple rules to comprehend and easily connect with real-time technologies. For the classification of different tasks, the DT algorithm is widely used [33].

Researchers derived many ML-based algorithms suitable for building IDS. Those ML-based methods such as k-NN, Support Vector SVM, LR, and DT have an influential role in detecting cyber attacks.

With all the advantages of DT algorithms and their ability to perform complex classification tasks, they pose problems related to computational complexity when dealing with high dimensional

features. In this paper, and to overcome these issues and build a high-accuracy model, we followed a methodology that places the classification of features in terms of importance in the first place and then selects them with a predefined threshold, furthermore due to the imbalanced dataset used in this work We proposed a hybrid data balancing methodology.

3. Research Methods

Data security and privacy are a big concern in today's world, and IDSs serve as the first line of defence for computer systems and networks. For the design of intelligent IDS, a variety of rule-based procedures or ML-based frameworks are used. In this work, a DT-based classifier is used to build our model. **Figure 1** shows in detail the proposed methodology:

1) Data collection and exploration.
2) Features selection.
3) Preprocessing.
4) Building and testing the model.

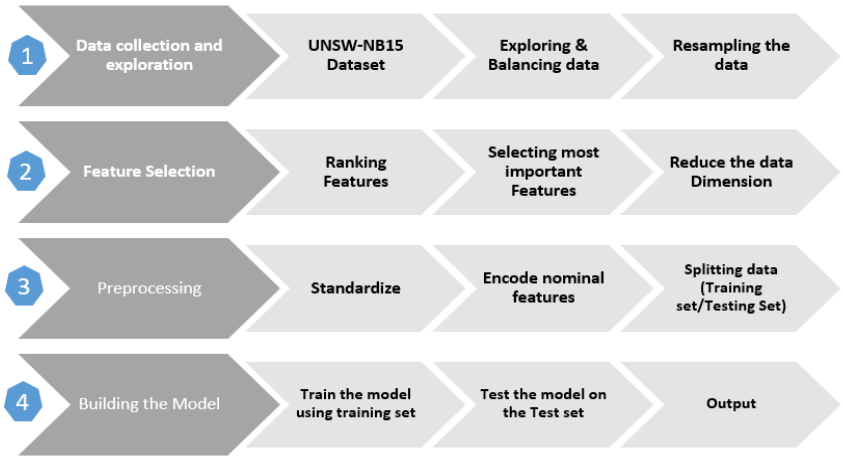

Figure 1. Intrusion detection model based on decision tree

3.1. Data collection and exploration

Security datasets are usually the collections of information records that include many security features and create a data-driven cybersecurity intrusion detection model. To detect malicious activity, it's crucial to grasp the essence of primary cybersecurity data and the dynamics of security threats [34], [35]. This paper uses an intrusion dataset known as "UNSW-NB 15".This dataset contains two class variables, normal and attack (0 for normal and 1 for attack). The UNSW-NB 15 [36] dataset was generated using the IXIA PerfectStorm tool in the Australian Centre for Cyber Security's (ACCS) Labs. The dataset contains 49 properties that determine the characteristics of each record. As seen in **Table 1**, the features type varies, with some being nominal, numeric, and others taking on time-stamp values [37]. The nominal features are proto, service, and state (coloured in blue see **Table 1**).

However, from the 49 security features, only 41 are relevant to our work. We will exclude attack_cat since it is out of the scope of the paper. Also, srcip, sport,dstip,dsport, Stime, and Ltime will be dropped as suggested by the dataset creators in [38].

Table 1. Security features of the UNSW-NB 15 datasets.

No.	Name	Type		No.	Name	Type
1	srcip	nominal		26	res_bdy_len	integer
2	sport	integer		27	Sjit	Float
3	dstip	nominal		28	Djit	Float
4	dsport	integer		29	Stime	Timestamp
5	proto	nominal		30	Ltime	Timestamp
6	state	nominal		31	Sintpkt	Float
7	dur	Float		32	Dintpkt	Float
8	sbytes	Integer		33	tcprtt	Float
9	dbytes	Integer		34	synack	Float
10	sttl	Integer		35	ackdat	Float
11	dttl	Integer		36	is_sm_ips_ports	Binary
12	sloss	Integer		37	ct_state_ttl	Integer
13	dloss	Integer		38	ct_flw_http_mthd	Integer
14	service	nominal		39	is_ftp_login	Binary
15	Sload	Float		40	ct_ftp_cmd	integer
16	Dload	Float		41	ct_srv_src	integer
17	Spkts	integer		42	ct_srv_dst	integer
18	Dpkts	integer		43	ct_dst_ltm	integer
19	swin	integer		44	ct_src_ ltm	integer
20	dwin	integer		45	ct_src_dport_ltm	integer
21	stcpb	integer		46	ct_dst_sport_ltm	integer
22	dtcpb	integer		47	ct_dst_src_ltm	integer
23	smeansz	integer		48	attack_cat	nominal
24	dmeansz	integer		49	Label	binary
25	trans_depth	integer				

3.2. Data Preparation

Data preparation (DP) is a challenging task in any ML project because each dataset is unique and tailored to the concerned project. Nonetheless, there are enough similarities amongst the predictive modelling projects to establish a loose sequence of phases and subtasks that will most likely be completed. This approach gives us a context in which we can think about the DP needed for the

project, which is informed by the project definition done before DP and the evaluation of ML algorithms was done after[39].

3.3. Data collection and exploration

A classification challenge known as an imbalanced classification problem occurs when the dissemination of samples across problem classes is uneven. The distribution can range from a bit of skew to a grave imbalance, with one instance in the minority class for hundreds, thousands, or millions in the majority class or classes [40]. In our case, the UNSW-NB 15 dataset is imbalanced, as shown in. Furthermore, 87.35% of the instance belongs to the class Normal versus 12.65% only for class Attacks, which means our dataset is highly imbalanced.

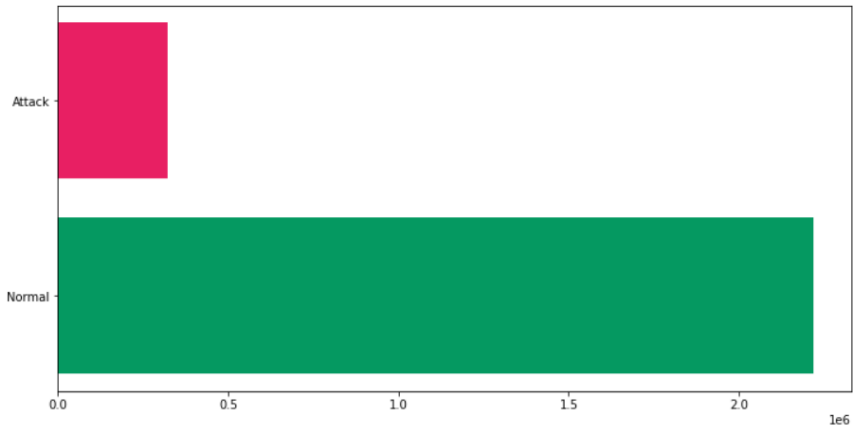

Figure 2. UNSW-NB 15 class distribution.

Because most ML algorithms for classification were created to assume the equal class distribution, CART's imbalanced learning is complex. Consequently, models will result in poor prediction accuracy, particularly for the minority class. This is not a surprise since the majority class is typically more abundant than the minority. So the issue is more susceptible to categorization errors for the minority class than for the majority class [41].

Conventionally, the Data-level approach and the Algorithm-level approach are two common approaches to deal with the imbalance learning tasks. The data-level approach methods adjust the class imbalance ratio to achieve a balanced class distribution. In the algorithm-level approach, classification algorithms are adjusted to improve the learning process, specifically the minority class [41]. However, the algorithmic level approach is ineffective when a high ratio of imbalanced classes exists. Contrarily, the data level method is preferred and encouraged in this scenario. The reason for this is that, depending on the situation, the data class composition can be modified to a "reasonably balanced" ratio by adding or removing many classes from the dataset [42], [43]. One of the most uncomplicated and most often utilized methods for the data level approach is resampling. Resampling techniques seek to increase the rate of minority samples. This can be achieved by eliminating samples from the majority class and adding examples from the minority class. Undersampling and oversampling are two approaches that show a good result, as seen in [44], [45]. Some frequent samples are deleted from the original dataset through undersampling until the desired balance ratio is obtained. The data to be deleted can be chosen at random or more effectively based on specific criteria, such as removing patterns located on the input space's outside regions [46].

Oversampling can be performed by increasing the number of minority class instances or samples by producing new examples or repeating some instances [47].

Random Undersampling (RUS) and Random oversampling (ROS) are the two main ways to resample an imbalanced dataset randomly. ROS lengthens the learning process, mainly if the original dataset has grown to be as large as ours but is as skewed. However, When the dataset size is small, this strategy is ideal. RUS is a form of data sampling that haphazardly chooses some majority class instances and withdraw them from the dataset until the aimed class propagation is attained [47], [48]. As NSW-NB15 Dataset is extremely large, the loss of some samples due to RUS should not be an issue as ample records will remain from which the DT classifier can be trained. Although either a RUS or ROS method can be efficient when employed alone on a dataset, experiments have shown that combining the two techniques can improve the overall model to fit the resampled dataset [40].

In this work, we focused on the data level approach, and we used the hybrid method by combining RUS and ROS. Although, we have applied ROS only on the training set to avoid duplication of samples in the testing set, as illustrated in

. This approach was recommended by [49] since the imbalance problem will not be solved by dividing the entire dataset into training and testing datasets using random sample techniques without adjusting the class distribution. As a result, the training data will have the same distribution as the original data set, which will be a problem throughout the learning process. **Figure 4** shows the dataset after applying the hybrid balancing approach.

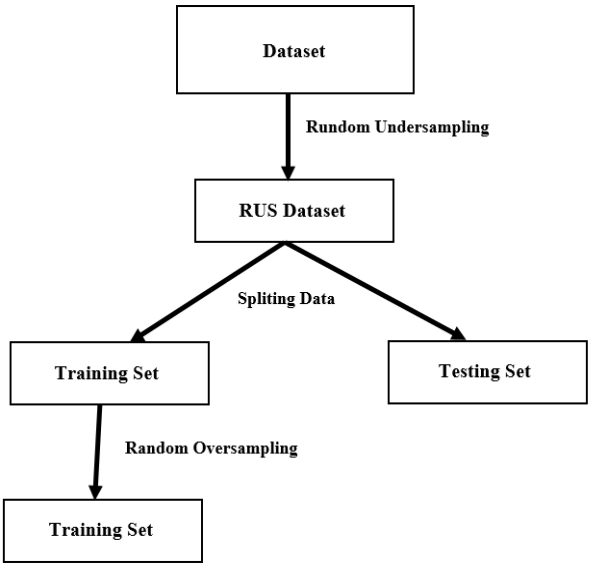

Figure 3. Our data balancing algorithm.

A train test dataset is a primary assessment method that divides the dataset into a train and a test dataset, then trains the learning model using the train data and measures performance using the test data [50]. In our case, we set 30% of the undersampled data for the testing and 70% for the training.

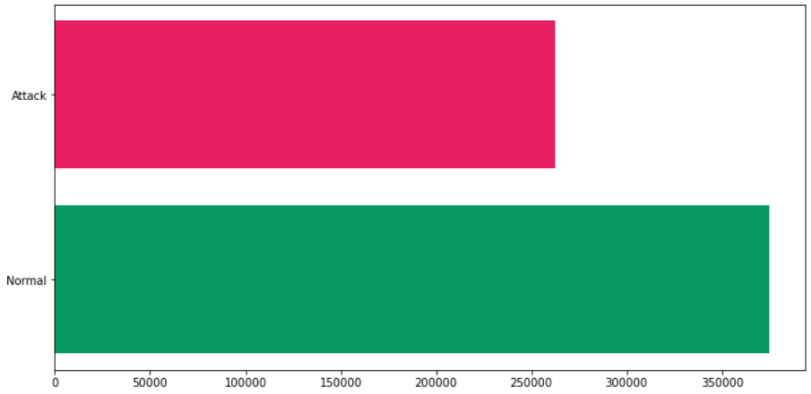

Figure 4. The Balanced Dataset.

3.4. Feature Encoding

The ML model's output is affected by its hyperparameters and processing of different variables. Encoding qualitative variables is mandatory since most ML models accept only numerical inputs. One Hot Encoding and Label Encoding are two approaches that can be used in this situation. However, One Hot Encoding will extend the feature space by generating new features [51]. As a result, we select Label Encoding to encode all the nominal features. Label encoding transforms labels into a numeric form so that ML algorithms can handle them. As we mentioned in section 3.1, nominal features are proto, service, and state.

3.5. Features Scaling

Feature scaling is a technique for normalizing a set of independent variables or data features. This can be achieved using two widely used methods, Normalization, and Standardization. Normalization (also known as Min-Max normalization) is a scaling technique in which the features are rescaled such that the data falls within the [0,1] range. Standardization is a scaling technique that rescales features to give them the properties of a standard normal distribution with a mean of 0 and a standard deviation of 1. The standardization algorithm was determined as shown in (1):

$$z = \frac{x - u}{\sigma} \tag{1}$$

Where Z is the new scaled value, μ is the mean of the distribution, and σ is the standard deviation of the distribution. Standardization centres around zero and scales the data point such that the norm is 0 and the standard deviation is 1.

3.6. Features Selection

FS is a technique for selecting the essential features from an ample feature space. In other words, it can choose among the features that are more important and irredundant for the classification problem [52]. ML-IDS models rely heavily on FS and ranking because removing the unnecessary features improves the detection accuracy while speeding up the computation and improving an IDS's overall efficiency [53]. Information Gain and the Gini index (GI) are two commonly used methods in this context. CART uses the GI, which is simple to implement, particularly for high dimensional features

[54]. The GI is an FS approach that determines how pure the features are concerning the class. The purity of a feature refers to how well it can discriminate between different groups [55].

If we consider dataset D, which includes samples from k different groups and P_i Denotes the probability of samples linked to class i at a given node. So the Gini impurity of the dataset D impurity is described as according to [56], [57] (2):

$$G(D) = 1 - \sum_{i=1}^{k} (P_i)^2 \tag{2}$$

We used Python and ITMO FS (Information Technologies, Mechanics and Optics University feature selection) library [58] to calculate the Gini index for all the security features. The lower the Gini, the more important features. After extracting the impurity features, we picked the top 11 (**Figure 5**) security features based on the selection threshold of 0.33 to construct our effective tree-based intrusion detection model.

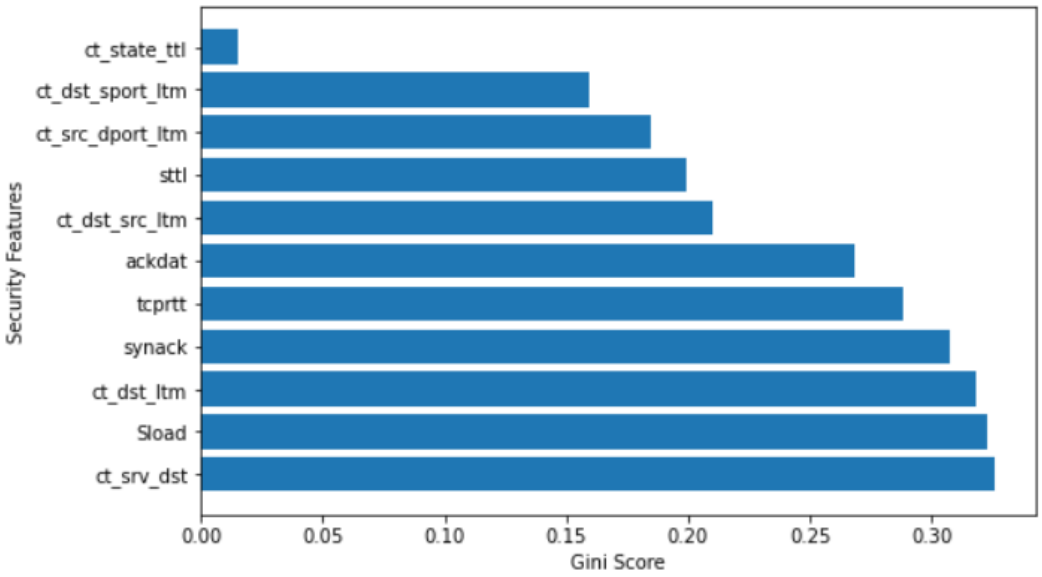

Figure 5. Selected features based on the threshold value of 0.33

3.7. Decision Tree Enhanced Intelligent Intrusion Detection Model

We have started by creating our E2IDS model once the security features are ready to process. Instead of using all the features available in the dataset, we used the selected features determined based on their Gini score and the defined threshold. The GI described in the previous section is used to identify the characteristic of the root node in each level. The feature with a lower Gini index is chosen as the root node. As a result, the tree-based model is expanded by adding the required number of internal branches, leaf nods, and connecting edges. Each tree leaf node is labelled with a target class, either attack or normal, and internal nodes are labelled with the security features selected earlier.

4. Experiments and Discussion

The experiments for this stud were conducted on a laptop with a CPU processor Intel Core i5-5300U with 2.30 GHz, 8 GB RAM, and a 64 bit Windows 10 operating system. Python programming language is used to implement the experiments. **Table 2** depicts the experimental implementation environment.

Table 2. Implementation environment.

Element	Description
Personal Computer	Windows 10, 2.30 GHz processor speed and 8 GB RAM
Programming Language	Python version 3.8
Dataset Management	Delimit 3.7 Open data files up to 2 billion rows and 2 million columns large

4.1. Performance Evaluation Measures

The results of experiments were assessed based on three measures. These measures are accuracy, sensitivity, and precision, that were computed as follows:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{3}$$

$$Recall\ (Sensitivity) = \frac{TP}{(TP + TN)} \tag{4}$$

$$Precision = \frac{TP}{(TP + FP)} \tag{5}$$

$$Fscore = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{6}$$

FP and FN are the numbers of false positives and negatives. TP and TN are the numbers of true positives and negatives. We also consider the ROC (Receiver Operating Characteristic) curve, which is formed by comparing the final security model's true positive rate (TPR) against the false positive rate (FPR).

4.2. Results and Comparisons

In this section, the results of the experiments were presented and compared with the results of the recent related works. Table 3 shows the E2IDS results in precision, recall, fscore, and accuracy for each class, either normal or attack.

Table 3. Result of experiments.

Class	Accuracy	Precision	Recall	Fscore
Normal	99%	99 %	99%	99%
Attack	98.3%	98.4%	98.4%	98.6%

In Table 4, the overall performance of the model is shown. The model showed improved performance for all metrics. Furthermore, the model's performance was benchmarked against state-of-the-art approaches.

Table 4. Overall Performance of the Model.

	Precision	Recall	Fscore	Support
Normal	99%	99%	99%	160,850
Attack	98.3%	98.4%	98.4%	96,177
Macro avg	98.7%	98.7%	98.7%	257,027
Weighted avg	98.8%	98.8%	98.8%	257,027

As shown in Table 5, the proposed model outperforms other models in all metrics and achieved an overall accuracy of 98.8%.

Table 5. Performance Comparison of the Proposed Model with other approaches.

	Precision	Recall	Fscore	Dataset Size	Features
Proposed Work	98.8%	98.8%	98.8%	2,540,047	11/41
Sarker et al., 2020	98.0%	98.1%	98.1%	25,193	14/41
Al-Omari et al., 2021	97.0%	97.0%	97.0%	175,341	19/41

However, since our dataset is highly imbalanced, we also used the Receiver operating characteristic (ROC) metric, an efficient tool to prove the overall accuracy of imbalance learning models. As shown in **Figure 6**, our model Area Under the Curve (AUC) score was 99%, indicating the model performance at distinguishing between the classes (Normal and Attack).

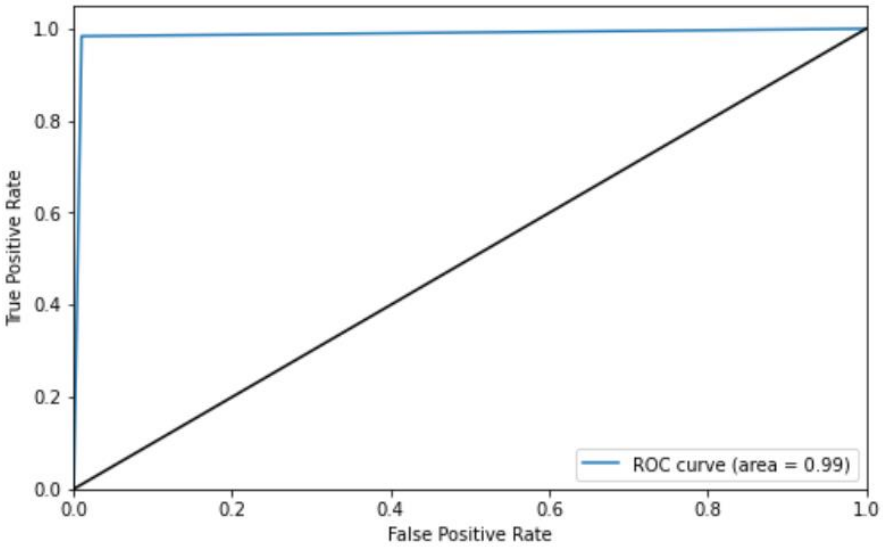

Figure 6. Receiver operating characteristic of the proposed model.

Additionally, compared to the other recent works, we used a larger dataset containing more than two million records and significantly reduce the feature space. We have observed that the E2IDS model outperforms the benchmark models. We first select the most important features before building the intrusion detection tree. As a result, it reduces model disparity and overfitting problems. We handled the imbalanced dataset problem using a hybrid method (RUS and ROS).

As a result, the model had improved prediction outcomes for unknown test instances while reducing the computing complexity. As a consequence of **Figure 7** and the previous experimental study, we can infer that our E2IDS model is more effective than existing intrusion detection models.

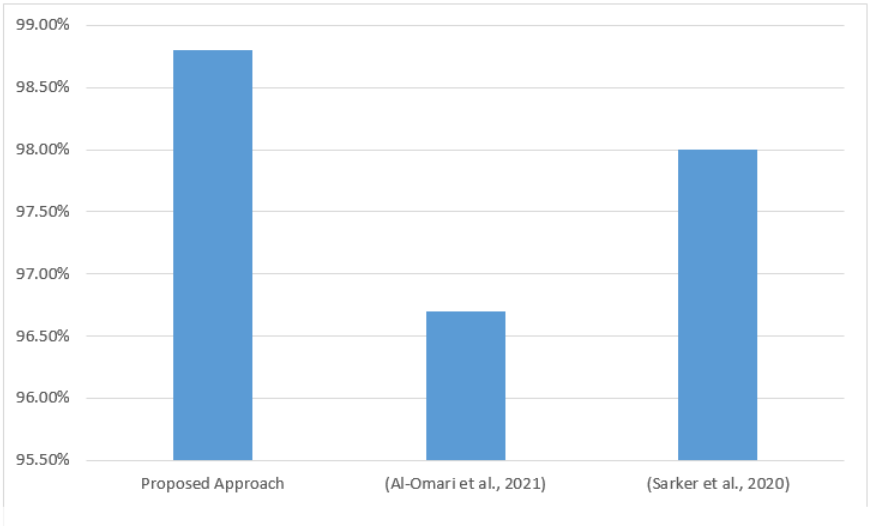

Figure 7. Visualizing the Accuracy of the Proposed Model against other Approaches.

5. Conclusion

Presently, there is an increase in cybercrimes as predators benefit from vulnerable systems. As a result, organizations need to improve their ability to identify, respond, and recover swiftly from a security incident. IDS is an efficient solution that can mitigate the intrusions of the computer environment by triggering alerts to speed up the incidence response process. Theoretically, IDS are based on the fact that an intruder's behaviour will be undoubtedly different from an authorized user's. This paper presented an enhanced intelligent intrusion detection model to detect a wide range of cyber-attacks. The model is built using an enhanced feature selection approach to select the most important features.

Additionally, the impurity of the security features was measured using the Gini Index. This approach enabled us to reduce the feature dimensions from 41 to 11. The model was generated using a sizeable imbalanced dataset, as shown in

. However, we proposed a hybrid method to balance the dataset. Finally, the performance of our model was proved using adequate metrics such as ROC, Precision, Fscore, and Recall. Finally, the evaluation results confirm that our model performs better than the recent similar works. Our future work will include expanding and applying our enhanced feature selection and data balancing methodologies to build a model that can predict the type of attacks.

Acknowledgement

The authors contributed equally to this paper. All authors have read and agreed to the published version of the manuscript. Funding Not applicable.

References

[1] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B. D. Payne, “Evaluating computer intrusion detection systems: A survey of common practices,” *ACM Comput. Surv.*, vol. 48, no. 1, 2015, doi: 10.1145/2808691.

[2] C. N. Modi and K. Acha, “Virtualization layer security challenges and intrusion detection / prevention systems in cloud computing : a comprehensive review,” *J. Supercomput.*, vol. 73, no. 3, pp. 1192–1234, 2017, doi: 10.1007/s11227-016-1805-9.

[3] R. M. Elbasiony, E. A. Sallam, T. E. Eltobely, and M. M. Fahmy, “A hybrid network intrusion detection framework based on random forests and weighted k-means Reda,” *Ain Shams Eng. J.*, vol. 4, no. 4, pp. 753–762, 2013, doi: 10.1016/j.asej.2013.01.003.

[4] K. Rai, M. S. Devi, and A. Guleria, “Decision Tree Based Algorithm for Intrusion Detection,” *Int. J. Adv. Netw. Appl.*, vol. 07, no. 04, pp. 2828–2834, 2016, [Online]. Available: https://www.researchgate.net/publication/298175900.

[5] J. Markey, “Information Security Reading Room for Intrusion Detection : A How-To Guide,” *SANS Inst.*, 2019.

[6] G. R. Jidiga and P. Sammulal, “Anomaly detection using machine learning with a case study,” in *Proceedings of 2014 IEEE International Conference on Advanced Communication, Control and Computing Technologies, ICACCCT 2014*, 2015, no. 978, pp. 1060–1065, doi: 10.1109/ICACCCT.2014.7019260.

[7] Y. Xin *et al.*, “Machine Learning and Deep Learning Methods for Cybersecurity,” *IEEE Access*, vol. 6, pp. 35365–35381, 2018, doi: 10.1109/ACCESS.2018.2836950.

[8] R. Jain and H. Shah, “An anomaly detection in smart cities modeled as wireless sensor network,” 2017, doi: 10.1109/ICONSIP.2016.7857445.

[9] C. Ioannou, V. Vassiliou, and C. Sergiou, “An Intrusion Detection System for Wireless Sensor Networks,” 2017, doi: 10.1109/ICT.2017.7998271.

[10] Q. Niyaz, W. Sun, A. Y. Javaid, and M. Alam, “A deep learning approach for network intrusion detection

system," 2015, doi: 10.4108/eai.3-12-2015.2262516.
[11] C. Yin, Y. Zhu, J. Fei, and X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017, doi: 10.1109/ACCESS.2017.2762418.
[12] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *J. Inf. Secur. Appl.*, vol. 50, p. 102419, 2020, doi: 10.1016/j.jisa.2019.102419.
[13] R. Zhao, Y. Yin, Y. Shi, and Z. Xue, "Intelligent intrusion detection based on federated learning aided long short-term memory," *Phys. Commun.*, vol. 42, p. 101157, 2020, doi: 10.1016/j.phycom.2020.101157.
[14] Z. Wang, Y. Liu, D. He, and S. Chan, "Intrusion detection methods based on integrated deep learning model," *Comput. Secur.*, vol. 103, 2021, doi: 10.1016/j.cose.2021.102177.
[15] G. Zhao, C. Zhang, and L. Zheng, "Intrusion detection using deep belief network and probabilistic neural network," *Proc. - 2017 IEEE Int. Conf. Comput. Sci. Eng. IEEE/IFIP Int. Conf. Embed. Ubiquitous Comput. CSE EUC 2017*, vol. 1, pp. 639–642, 2017, doi: 10.1109/CSE-EUC.2017.119.
[16] B. M. Serinelli, A. Collen, and N. A. Nijdam, "Training guidance with KDD Cup 1999 and NSL-KDD data sets of ANIDINR: Anomaly-based network intrusion detection system," *Procedia Comput. Sci.*, vol. 175, no. 2019, pp. 560–565, 2020, doi: 10.1016/j.procs.2020.07.080.
[17] I. H. Sarker and A. S. M. Kayes, "ABC-RuleMiner: User behavioral rule-based machine learning method for context-aware intelligent services," *J. Netw. Comput. Appl.*, vol. 168, no. September 2019, p. 102762, 2020, doi: 10.1016/j.jnca.2020.102762.
[18] F. Amiri, M. Rezaei Yousefi, C. Lucas, A. Shakery, and N. Yazdani, "Mutual information-based feature selection for intrusion detection systems," *J. Netw. Comput. Appl.*, vol. 34, no. 4, pp. 1184–1199, 2011, doi: 10.1016/j.jnca.2011.01.002.
[19] J. Gu and S. Lu, "An effective intrusion detection approach using SVM with naïve Bayes feature embedding," *Comput. Secur.*, vol. 103, p. 102158, 2021, doi: 10.1016/j.cose.2020.102158.
[20] S. Garg and S. Batra, "Fuzzified Cuckoo based Clustering Technique for Network Anomaly Detection," *Comput. Electr. Eng.*, vol. 71, pp. 798–817, 2018, doi: 10.1016/j.compeleceng.2017.07.008.
[21] F. Kuang, W. Xu, and S. Zhang, "A novel hybrid KPCA and SVM with GA model for intrusion detection," *Appl. Soft Comput. J.*, vol. 18, pp. 178–184, 2014, doi: 10.1016/j.asoc.2014.01.028.
[22] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, "Building an intrusion detection system using a filter-based feature selection algorithm," *IEEE Trans. Comput.*, vol. 65, no. 10, pp. 2986–2998, 2016, doi: 10.1109/TC.2016.2519914.
[23] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, and K. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 424–430, 2012, doi: 10.1016/j.eswa.2011.07.032.
[24] S. H. Kang and K. J. Kim, "A feature selection approach to find optimal feature subsets for the network intrusion detection system," *Cluster Comput.*, vol. 19, no. 1, pp. 325–333, 2016, doi: 10.1007/s10586-015-0527-8.
[25] S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, and H. Karimipour, "Cyber intrusion detection by combined feature selection algorithm," *J. Inf. Secur. Appl.*, vol. 44, pp. 80–88, 2019, doi: 10.1016/j.jisa.2018.11.007.
[26] I. H. Sarker, A. Colman, J. Han, A. I. Khan, Y. B. Abushark, and K. Salah, "BehavDT: A Behavioral Decision Tree Learning to Build User-Centric Context-Aware Predictive Model," *Mob. Networks Appl.*, vol. 25, no. 3, pp. 1151–1161, 2020, doi: 10.1007/s11036-019-01443-z.
[27] I. H. Sarker, Y. B. Abushark, F. Alsolami, and A. I. Khan, "IntruDTree: A machine learning based cyber security intrusion detection model," *Symmetry (Basel).*, vol. 12, no. 5, pp. 1–15, 2020, doi: 10.3390/SYM12050754.
[28] M. Aloqaily, S. Otoum, I. Al Ridhawi, and Y. Jararweh, "An intrusion detection system for connected vehicles in smart cities," *Ad Hoc Networks*, vol. 90, p. 101842, 2019, doi: 10.1016/j.adhoc.2019.02.001.
[29] A. S. Eesa, Z. Orman, and A. M. A. Brifcani, "A new feature selection model based on ID3 and bees algorithm for intrusion detection system," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 23, no. 2, pp. 615–622, 2015, doi: 10.3906/elk-1302-53.
[30] S. Puthran and K. Shah, "Intrusion Detection Using Improved Decision Tree Algorithm with Binary and Quad Split," in *Security in Computing and Communications*, 2016, pp. 427–438.
[31] M. Al-Omari, M. Rawashdeh, F. Qutaishat, M. Alshira'H, and N. Ababneh, "An Intelligent Tree-Based Intrusion Detection Model for Cyber Security," *J. Netw. Syst. Manag.*, vol. 29, no. 2, pp. 1–18, 2021, doi: 10.1007/s10922-021-09591-y.
[32] B. Ingre, A. Yadav, and A. K. Soni, "Decision Tree Based Intrusion Detection System for NSL-KDD

Dataset," in *Information and Communication Technology for Intelligent Systems (ICTIS 2017) - Volume 2*, 2018, pp. 207–218.
[33] T. Efraim, R. Sharda, and D. Delen, "Business intelligence and analytics: Systems for decision support." Pearson Higher Ed, 2014.
[34] I. Amit, J. Matherly, W. Hewlett, Z. Xu, Y. Meshi, and Y. Weinberger, "Machine learning in cyber-security - problems, challenges and data sets," *arXiv*, 2018.
[35] I. H. Sarker, A. S. M. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, "Cybersecurity data science: an overview from machine learning perspective," *J. Big Data*, vol. 7, no. 1, 2020, doi: 10.1186/s40537-020-00318-5.
[36] Australian Centre for Cyber Security (ACCS), "UNSW_NB15," *Kaggle*. .
[37] N. Moustafa and J. Slay, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Inf. Secur. J.*, vol. 25, no. 1–3, pp. 18–31, 2016, doi: 10.1080/19393555.2015.1125974.
[38] "The UNSW-NB15 Dataset | UNSW Research." https://research.unsw.edu.au/projects/unsw-nb15-dataset (accessed Sep. 24, 2021).
[39] J. Brownlee, "Data Preparation for Machine Learning: Data Cleaning, Feature Selection, and Data Transforms in Python," p. 398, 2020.
[40] J. Brownlee, "Imbalanced Classification with Python," *Mach. Learn. Mastery*, p. 463, 2020.
[41] A. Ali, S. M. Shamsuddin, and A. L. Ralescu, "Classification with class imbalance problem: A review," *Int. J. Adv. Soft Comput. its Appl.*, vol. 7, no. 3, pp. 176–204, 2015.
[42] D. Zhang, W. Liu, X. Gong, and H. Jin, "A novel improved SMOTE resampling algorithm based on fractal," *J. Comput. Inf. Syst.*, vol. 7, no. 6, pp. 2204–2211, 2011.
[43] Y. Pristyanto, I. Pratama, and A. F. Nugraha, "Data level approach for imbalanced class handling on educational data mining multiclass classification," *2018 Int. Conf. Inf. Commun. Technol. ICOIACT 2018*, vol. 2018-Janua, pp. 310–314, 2018, doi: 10.1109/ICOIACT.2018.8350792.
[44] S. Cateni, V. Colla, and M. Vannucci, "A method for resampling imbalanced datasets in binary classification tasks for real-world problems," *Neurocomputing*, vol. 135, pp. 32–41, 2014, doi: 10.1016/j.neucom.2013.05.059.
[45] T. Khoshgoftaar, C. Seiffert, and J. Van Hulse, "Hybrid sampling for imbalanced data," in *Proceedings of IRI*, 2008, vol. 8, pp. 202–207.
[46] N. V Chawla, "C4. 5 and imbalanced data sets: investigating the effect of sampling method, probabilistic estimate, and decision tree structure," in *Proceedings of the ICML*, 2003, vol. 3, p. 66.
[47] R. Mohammed, J. Rawashdeh, and M. Abdullah, "Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results," *2020 11th Int. Conf. Inf. Commun. Syst. ICICS 2020*, no. April, pp. 243–248, 2020, doi: 10.1109/ICICS49469.2020.239556.
[48] J. Prusa, T. M. Khoshgoftaar, D. J. DIttman, and A. Napolitano, "Using Random Undersampling to Alleviate Class Imbalance on Tweet Sentiment Data," *Proc. - 2015 IEEE 16th Int. Conf. Inf. Reuse Integr. IRI 2015*, pp. 197–202, 2015, doi: 10.1109/IRI.2015.39.
[49] E. AT, A. M, A.-M. F, and S. M, "Classification of Imbalance Data using Tomek Link (T-Link) Combined with Random Under-sampling (RUS) as a Data Reduction Method," *Glob. J. Technol. Optim.*, vol. 01, no. S1, pp. 1–11, 2016, doi: 10.4172/2229-8711.s1111.
[50] M. B. Structures, *Big Data Analytics for Intelligent Healthcare Management*. Elsevier, 2019.
[51] A. Zheng and A. Casari, *Feature engineering for machine learning*, no. September. 2018.
[52] A. O. Balogun, S. Basri, S. J. Abdulkadir, and A. S. Hashim, "Performance analysis of feature selection methods in software defect prediction: A search method approach," *Appl. Sci.*, vol. 9, no. 13, 2019, doi: 10.3390/app9132764.
[53] S. Mukkamala and A. H. Sung, "Feature ranking and selection for intrusion detection systems using support vector machines," in *Proceedings of the second digital forensic research workshop*, 2002, pp. 1–10.
[54] T. Thomas, A. P. Vijayaraghavan, and S. Emmanuel, *Machine learning approaches in cyber security analytics*. 2019.
[55] O. Al-Harbi, "A Comparative Study of Feature Selection Methods for Dialectal Arabic Sentiment Classification Using Support Vector Machine," vol. 19, no. 1, pp. 167–176, 2019, [Online]. Available: http://arxiv.org/abs/1902.06242.
[56] S. Lefkovits and L. Lefkovits, "Gabor Feature Selection Based on Information Gain," *Procedia Eng.*, vol. 181, pp. 892–898, 2017, doi: 10.1016/j.proeng.2017.02.482.
[57] V. Jain, A. Phophalia, and J. S. Bhatt, "Investigation of a Joint Splitting Criteria for Decision Tree Classifier Use of Information Gain and Gini Index," *IEEE Reg. 10 Annu. Int. Conf.*

Proceedings/TENCON, vol. 2018-Octob, no. October, pp. 2187–2192, 2019, doi: 10.1109/TENCON.2018.8650485.

[58] N. Pilnenskiy and I. Smetannikov, "Feature selection algorithms as one of the python data analytical tools," *Futur. Internet*, vol. 12, no. 3, 2020, doi: 10.3390/fi12030054.