Cluster Comput (2019) 22:S10549–S10565
https://doi.org/10.1007/s10586-017-1109-8

An enhanced J48 classification algorithm for the anomaly intrusion detection systems

Shadi Aljawarneh[1] · **Muneer Bani Yassein**[1] · **Mohammed Aljundi**[1]

Received: 13 April 2017 / Revised: 21 June 2017 / Accepted: 10 August 2017 / Published online: 5 September 2017

Abstract In this paper, we have developed an enhanced J48 algorithm, which uses the J48 algorithm for improving the detection accuracy and the performance of the novel IDS technique. This enhanced J48 algorithm is seen to help in an effective detection of probable attacks which could jeopardise the network confidentiality. For this purpose, the researchers used many datasets by integrating different approaches like the J48, Naive Bayes, Random Tree and the NB-Tree. An NSL KDD intrusion dataset was applied while carrying out all experiments. This dataset was divided into 2 datasets, i.e., training and testing, which was based on the data processing. Thereafter, a feature selection method based on the WEKA application was used for evaluating the efficacy of all the features. The results obtained suggest that this algorithm showed a better, accurate and more efficient performance without using the above-mentioned features when compared to the feature selection procedure. An implementation of this algorithm guaranteed the dataset classification based on a detection accuracy of 99.88% for all the features when using the 10-fold cross-validation test, a 90.01% accuracy for the supplied test set after using the complete test datasets along with all the features and a 76.23% accuracy for supplying the test set after using the test-21 dataset along with all features.

✉ Shadi Aljawarneh
saaljawarneh@just.edu.jo

Muneer Bani Yassein
masadeh@just.edu.jo

Mohammed Aljundi
mohammed.aljundi@yahoo.com

[1] Faculty of Computer and Information Technology, Jordan University of Science and Technology, Irbid, Jordan

Keywords J48 · IDS · Feature selection · Fold cross-validation · Weka · NSL_KDD

1 Introduction

The intrusion detection systems (IDS) are defined as efficient security tools which are used for improving the security of the communicating and the information systems, as they primarily focus on detecting malicious network traffics. An IDS is seen to be very similar to many processes like firewalls, antivirus software, and can access the control schemes. The IDS is classified depending on detection as the signature detection and anomaly detection system. For the signature-based detection systems, the systems identify the traffic pattern or the application data as malicious and this requires an updated database for storing all the new attack signatures, whereas the anomaly detection system compares all activities against the normal defined behaviour [1].

The main objective of the IDS system is detecting and then raising an alarm if the network is attacked. The best IDS process detects the new or more malicious attacks within a short time period and carries out the necessary actions. The currently used IDS systems do not show 100% accuracy, hence, this study has been carried out for improving and increasing the IDS system accuracy [2]. Many of the machine learning techniques have been used for helping in the detection of the network attacks, improving the accuracy detection rate and developing effective classification and clustering models for distinguishing between a normal and an abnormal behaviour packet. The procedure of detecting the intrusion accurately from the complete network traffic is classified as the classification problem.

The IDS systems are classified as per the detection methods used for identifying all the malicious attacks [3]. A

misuse or a signature detection technique identifies signatures or patterns present in the existing attacks within the network traffic. This misuse detection system requires an updated database for storing the new attack signatures. But, new attacks are not detected until the system gets trained. The anomaly detection uses an approach based on detection of the traffic anomalies by identifying the behaviour which is different from a normal behaviour. Hence, this shows that though IDS can handle new attacks, it is unable to detect or identify a particular attack pattern [4].

Several researchers have used the Data mining for improving the IDS by offering an external intrusion detection that identifies the presence of any existing boundaries within a normal network activity. This helps in distinguishing between a normal and an abnormal activity [5]. The Data Mining is also applied in the IDS for identifying if any attacks are present in the system, improving the detection accuracy rate and developing effective classification and clustering models for distinguishing between a normal and an abnormal behaviour packet. The procedure of detecting the intrusion accurately from the complete network traffic is classified as the classification problem.

The classification models help in identifying any malicious attacks, improving the accuracy detection rate and decreasing false alarms. Many data mining algorithms have been proposed earlier for generating an efficient IDS like the Naïve Bayes [6], AdaBoost [7], J48 Decision Tree [8], ANN [9], Support Vector Machines [10] etc. All these algorithms are integrated with the different models for distinguishing between the malicious attacks and determining a normal behaviour for detecting unknown malicious attacks. The various IDSs are classified depending on their detection technique, architecture along with their post-detection activity [11].

In this study, the researchers have worked towards improving the J48 algorithm for decreasing the human efforts in handling the increasing and ever-changing intruder attacks and providing a network protection against internal and external attacks; along with improving the accuracy detection rate and the performance of the anomaly IDSs. This enhanced J48 algorithm shows a better performance as compared to the other systems and algorithms, due to the following reasons:

- The enhanced J48 algorithm helps in enhancing the detection attacks in the anomaly IDS.
- The enhanced J48 algorithm shows better detection accuracy as compared to many other algorithms.
- This enhanced J48 algorithm is thought to show a better value for TP, TN, FP and FN accuracy percent values as compared to many other techniques.

The currently used IDS techniques used for detecting the unauthorised attacks do not show 100% accuracy. Thus, improving the application of the J48 algorithm was important for improving the performance and the accuracy of detecting any probable attacks. Hence, in this study, the researchers have aimed to improve the J48 classification algorithm for enhancing the IDS performance. This research is important as it would help in preventing an illegal or unethical intrusion in a network. An effective system helps in an accurate detection of any probable attacks which increased the risk of these attacks penetrating the network system and accessing the data.

The remaining paper is organised in the following manner: Sect. 2 presents an overview of the various data mining techniques, where the J48 technique has been described. In Sect. 3, a literature review has been presented for intrusion detection by different techniques. In Sect. 4, the researchers have described the proposed algorithm and its implementation. Also, an experimental dataset and a simulation environment have been described here. Section 5 presents all the experiments and their results, while Sect. 6 describes the conclusion of this work and future work.

2 An overview of the various data mining techniques

The advanced IDS methods make use of various data mining processes along with different knowledge bases for their training in the detection of unknown attacks or abnormal behaviours. The IDS consists of different data mining techniques, methods and algorithms for detecting the system attacks, thus helping the system to detect the intrusions more dynamically [4]. Using the data mining techniques in the IDS helps in improving the system performance and the security, and the systems can detect known or unknown attacks. The data mining help in the network intrusion detection problems by: (i) Processing and analysing huge amounts of data, (ii) effectively discovering the unknown data and (iii) improving the security analysis by carrying out data visualisation and summarization.

Many data mining processes were developed for the intrusion detection. Several researchers used unique approaches for improving the data classification accuracy. In this study, the researchers have applied many data mining techniques like the NBTree, RandomTree, RandomForest, support vector machine (SVM), artificial neural network (ANN) etc. [12].

The decision tree is a very popular and widely used classification algorithm, which has the following characteristics:

- The decision tree algorithm or the iterative dichotomiser (ID3) has been popular since the 1970s.
- A classification and regression tree (CART) is used for developing binary decision trees, as described earlier [13].

```
1: Create a root node N;
2: IF (T belongs to same category C)
       {leaf node = N;
          Mark N as class C;
          Return N;
        }
3: For i=1 to n
       {Calculate Information_gain  (Ai);}
4: ta= testing attribute;
5: N.ta = attribute having highest information_gain;
6: if (N.ta == continuous )
        { find threshold;}
7: For (Each T in splitting of T)
8:       if (T is empty)
            {child of N is a leaf node;}
           else
            {child of N= dtree T)}
10: calculate classification error rate of node N;
11: return N;
```

Fig. 1 The pseudo code for the C4.5 (J48) algorithm

- Later, Quinlan [14,15] proposed a C4.5 algorithm, which is now used as a benchmark system which is compared to the newly developed supervised learning algorithms.

The ID3, CART, and the C4.5 are a form of a greedy technique that is a top-down recursive divide-and-conquer form of approach [2]. Furthermore, ID3 and C4.5 techniques are slower than some of the other types of decision tree algorithms, however, they are able to handle the continuous attributes/features and tackle the missing values [16].

2.1 J48 algorithm

C4.5/J48 is a widely used machine learning algorithm, which is a decision tree algorithm. This is a type of the ID3 algorithm, developed by Quinlan [14] and is described in Fig. 1.

The C4.5/J48 algorithm differs from the IDE3 as while building a decision tree, the algorithm can accept the continuous and the categorical attributes. Because of a high noise or a very detailed training data set, the J48 algorithm uses an enhanced technique of tree pruning for decreasing the misclassification error. Furthermore, this algorithm also used a greedy divide-and-conquer method for recursively inducing decision trees containing the database/dataset attributes for further classification. In any decision tree, classification is a major performance parameter. The classification error can be defined as the percentage of the misclassified cases [17]. The C4.5 algorithm is seen to accept the continuous and the categorical attributes while developing a decision tree. This decision tree can be developed by making use of the top down or the bottom up approach. Furthermore, the J48 classifier algorithm is divided into a dataset based on the different attribute values of the present data for separating a probable prediction. The decision tree contains many decision nodes and leaf nodes, wherein the decision nodes determine the test of the attributes while the leaf nodes represent the class values [18]. Every path in the decision trees from the root to the leaf node determines the rule. This J48 classifier algorithm can develop its decision tree depending on the information of the theoretical attribute values of the present training data. Also, in the case of a J48 algorithm, every feature or attribute separately estimates the gain value and the calculation process is continued till the prediction process is completed. An appropriate feature is defined as the feature which gives a lot of information regarding the data instances. This feature can be classified as a root node if it consists of the maximal information gain. After selecting the root node, the J48 algorithm can divide the training data into many subsets which correspond to the various values of a chosen feature and this process is repeated for every subset till every subset is assigned to one class.

The J48 algorithm consists of many features described below:

- It is accessible as an open source in the WEKA interface in Java.
- The algorithm helps in building easy to understand models.
- The algorithm makes use of the categorical and the continuous values.
- The algorithm provides a technique known as the imputation, which deals with missing values. This technique helps in resolving the missing value problem, which is a significant feature, after determining the missing values based on the available data.
- The algorithm also provides a tree pruning process, which helps in building small trees and avoiding over-fitting of the data.
- Also, the algorithm provides the subtree replacement process which decreases the classification error after replacing the subtree with a leaf.

3 Related work

Many studies published earlier have focused on investigating the effect of applying the J48 for enhancing the accuracy of the intrusion detection. The main keywords used for searching include J48, intrusion detection system, detection, security and algorithms.

Several IDSs use a single algorithm system which classifies data as either anomalous or normal. But, using one classifying system is unable to provide a precise detection system which detects and reports intrusions with a low rate of false alarms. Panda et al. [19] stated that integrating a hybrid intelligent scheme, which needed different classifiers to be implemented, would improve the detection and make

it very genuine, thereby improving the result quality. In this paper, the researchers have applied a 2-class classification strategy which is based on the 10-fold cross validation process, which would increase the rate of intrusion detection and also decrease the rate of false alarms [19]. This section highlights the important outcomes of various published studies and discusses the limitations and strengths of the systems used by them.

A rapid advancement in the field of information technology has introduced many machine learning techniques that can be implemented in the IDSs. Aburomman and Reaz [20] carried out a study which described the different algorithms used for classifying the intrusions based on a popular machine learning method. They studied different homogeneous or heterogeneous systems along with various hybrid techniques. They stated that implementing the ensemble-based techniques helped in solving the pattern classification-based problems [20].

After reviewing the available literature, the researchers concluded that several methods could be applied that used various classifiers. For example, some approaches decreased the variance and included boosting and bagging, whereas some decreased the bias. Some other methods, like cascading helped in developing new attributes. In such attributes, every classifier could handle a specific data set, whereas the rest was handled by other classifiers observed within the whole ensemble [20].

Moreover, the authors also considered many ensemble methods which depended on the voting system as they were considered to be simple processes that could generate the desired outcome. Many studies have stated that the hybrid method is popularly used for detection of malicious activities within the network. This method needs to integrate the feature selection and use one classifier. The voting system is very popular amongst the techniques for combining the classifiers. This system is very reliable because it is able to correct all the errors produced by the other classifiers, thus, improving the performance of the classifiers [20]. These authors constructed a system containing the bagging and the boosting ensembles, which also integrated 4 other conventional algorithms, like the J48 (decision trees), Bayes, IBK (nearest neighbour) and the JRip (rule induction). They developed heterogeneous ensembles after using the stacking strategy. In their study, they integrated each of the above-mentioned four algorithms for carrying out a meta-level classification system. This approach showed 60% accuracy. Their results showed that their heterogeneous ensembles, created using the bagging and the boosting processes, significantly improved the accuracy rate. Furthermore, their approach also showed 90% accuracy while detecting the known intrusions. On the other hand, this heterogeneous ensemble designed using a stacking strategy showed a significant decrease (46.84%) in the false positive [20].

The results described by Aburomman and Reaz [20] contradicted those shown by Panda et al. [19]. In this study, the researchers have noted that using the J48 decision tree showed a low detection rate of 90.7%. Also, using the J48 along with the radial basis function (RBF) neural showed a higher false alarm rate of 5.6%.

In another study, Goeschel [21] aimed to identify the false positive rate by developing a system that used the data mining processes for decreasing the false positives. Their technique combined the Naïve Bayes, SVM, and the decision trees; which would improve the efficiency and the accuracy of their IDS. The first step showed that SVM was used as the new binary model for classifying if the traffic was an attack or not. After identifying the abnormal attacks, Step 2 comprised of transferring the attacking traffic using a decision tree. In this step, the probable attacks were transferred using the decision trees after integrating the J48 algorithm and also excluding any irrelevant labels. The decision tree was seen to be a versatile system which could detect true positives or also detect if the alarm was new to their system. Also, this system raised a true positive alarm if it detected a leaf. But, if no leaf was detected, then the system would tag the alarm as new. In Step 3, the decision tree and the Naïve Bayes worked collectively for detecting the previously undetected attacks. This system was very effective as it showed an overall accuracy of 99.62% and a relatively lower false positive rate of 1.57 %. Though the system was seen to be very efficient, the authors suggested that further research was to be carried out for the implementation of this model on many other network systems [21].

The problem of network data protection was reviewed in a study by Sharma and Gupta [22]. An increasing use of the network services and an increase in the number of intrusions and system attacks has led to the introduction of several problems related to the intrusion detection. In their study, the authors studied the intrusion detection using the various data mining processes and they highlighted the difficulties observed in the IDSs, which arose from many sources used by the IDS for analysing the data. Several IDSs classified the data using the anomaly detection or the misuse detection. Every approach had some advantages and some limitations. However, it must be mentioned that no IDS can modify all the problems within any system. The best detection system is one which provides a more acceptable security level, which is achieved by improving the efficiency of the detection of the bigger intrusion attacks. Hence, the authors have suggested that a perfect IDS process can provide a more accurate confidence rate for the results seen, which is an important measure for any IDS. As per many review articles, using the J48 decision tree provided promising insights while improving the IDS performance [22].

Some threats faced by the virtualised systems are common threats that are encountered by any system, as they affect

all the computerised systems, and include the DoS attacks and the denial-of-service attacks. However, other threats or vulnerabilities are more specific for the virtual machines. Additionally, some VM vulnerabilities occur due to a vulnerability which is seen in one of the VM systems which is then extended to other VM systems. They could also harm the systems in some cases. This is possible as a majority of the VMs use similar physical hardware systems [23].

For improving the attack detection accuracy in the network systems, many researchers have focused on designing and developing some machine learning algorithms which are integrated within the algorithms. In their study, Noureldien and Yousif [24] identified a major abnormal attack within their network traffic, called as the denial of service (DoS) attack. The DoS attack comprised of many forms of attacks like the Teardrop, Smurf, Land, Neptune and Back. The authors investigated the accuracy of the algorithms which were used for detecting the DoS attacks. All these algorithms were used in different systematic techniques and included Logistic, IBK, PART, J48, InputMapped, BayesNet, and Random Committee. The authors used NSL-KDD dataset as the experimental tool, while WEKA was the mining tool. Their results showed that the best algorithm for detecting the Smurf attack was the Random Committee, and it showed 98.62% accuracy, while PART was the best algorithm for detecting the Neptune attack and it showed 98.55% accuracy. PART and J48 showed a similar accuracy for detecting the Smurf attack, with a very slight difference in accuracy values in comparison to those shown by the Random Committee algorithm. Also, PART had an average DoS attack detection rate. On the other hand, InputMapped was the worst algorithm for detecting the DoS attacks.

Abdeljalil and Mara [25] tested the performance of 3 of the machine learning algorithms, i.e., Decision Tree J48, SVM, and Neural Network. They tested the performance for intrusion detection using the parameters of detection rate, accuracy, and false alarm rate. Their results suggested that the J48 algorithm performed much better than the SVM and the Neural Network [24] and the authors showed that the J48 algorithm was better with a low false positive rate and high true positive rate [24]. These results were similar to those obtained earlier [26].

Also, in another study, Mazraeh et al. [27], the authors showed the effectiveness of using the Decision Tree J48 algorithm for intrusion detection. They designed a model which was based on some algorithm related to this decision tree and contained several values. The advantage of using a decision tree algorithm was that it effectively helped in data interpretation. Also, the high system efficiency was dependent on the properties selected. An improvement in the properties further increased the total cost of this proposed system. The decision tree algorithm is classified into 2 groups, i.e., classification and regression tree (CART) and the C4.5. There was an improvement in the performance of the decision tree J48 when it was integrated with the C4.5. For improving the accuracy of intrusion detection, the researchers suggested using a policy for identifying the different forms of the system and identifying the requirements for its supervision. This improved the accuracy of the positive and the negative alarms. These alarms indicated that the warnings were not released in an accurate manner which matched their needs for identifying the intrusions. In their study, the authors used 3 different learning algorithms, i.e., the SVM, J48 and the Naïve Bayes, and evaluated their efficiencies. Their results showed that the J48 algorithm was very effective, in comparison to the other algorithms, and had 97% efficiency and 91.8% average accuracy value. Hence, their results showed that the Decision tree J48 algorithm was very effective in detecting intrusions [27].

For an accurate identification and detection of the intrusions within any computerised network, Gaikwad et al. [28] studied the effectiveness of an IDS with the help of bagging using the partial decision TreeBase classifier. They proposed this technology because of a higher false alarm rate and a lower accuracy noted in the conventional IDSs. The bagging ensemble process is very popular in the IDS because of its ease in the partial decision tree classifier. The authors selected some relevant features based on the risks presented by every type of attack. This helped in significantly improving the classifier accuracy. The system designed by Gaikwad et al. [28] was evaluated for many factors like true positives, false positives, and the system accuracy. The system accuracy was determined using a cross-validation study, wherein the system showed 99.71% accuracy, with its classification accuracy higher than all the other classifiers.

Studying the digital information system is very important in today's day and age due to a high risk observed because of unethical attacks. Identifying the threats using an IDS is a very challenging problem. In their study, Nema et al. [29] tried to improve the accuracy for intrusion detection using the Layered Approach with help of an SVM having Feature Reduction. They attempted to identify several attacks like the probe, U2R, DoS and R2L. Their results provided a promising insight about the implementation of the SVM by integrating the genetic approach. The SVM is very commonly applied in machine learning after integrating some learning algorithms which allow data analysis and identification of specific patterns. Also, the SVMs need data classification using different class labels which helped in determining many supporting vectors. Furthermore, the SVM is seen to provide a very generic mechanism which used the kernel function and included the linear, polynomial or the sigmoid function [29].

In another study, Onik et al. [3] carried out a comparative analysis after integrating the feature selection approach and using the data mining tool known as WEKA, which

integrated the J48 feature selection. They studied the effectiveness of the J48 performance using various filter processes. Based on the application of the J48 feature selection, the authors could conclude that the J48 showed an enhanced prediction time and also lowered the computational time. Using the feature selection further decreased the redundant and the irrelevant features and also improved the representation of the optimal features. Moreover, the feature selection decreased the data redundancy by removing the irrelevant data and lowered the time complexity noted in many IDSs. Many researchers have stated that the SVM was a very appropriate method for determining the appropriate feature subset and detecting any probable attack. In their study, Onik et al. [3] studied the filter method using different search techniques. They finally used the J48 as the feature classifier in their IDS model and their results showed that every feature approach consisted of an optimal feature which differed from others. Furthermore, the authors also concluded that using the J48 helped in improving the quality of the features which were developed. The authors also used an analytical comparison approach along with many feature selection techniques, which integrated the J48 classification tree. Use of a decision tree was considered to be very effective in any IDS. Hence, the authors studied many methods for improving the performance of the J48 by decreasing the redundant features. Also, the filter feature extraction process in the data mining and its use in the J48 classifier showed some positive results [3].

Determination of abnormal activities resulting due to anomaly intrusion was investigated using different algorithms and processes. These processes made use of the true positive and the false positive parameters and compared the performance of the techniques. In one study, Modi et al. [30], used the testing dataset (i.e., KDD-CUP-99), which used a large amount of data which needed a pre-processing technique. In their study, the authors proposed a technique which could be applied for feature selection along with feature elimination. This helped in decreasing the number of the relevant features and also aided in selecting an appropriate subset of classifiers which would provide proper classification techniques along with good multi-classifier models. The authors studied the use of some classifiers like the Naïve Bayes, J48, and the SVM. Out of these, the J48 classification algorithm was a type of the source classifier belonging to the C4.5 algorithms. These C4.5 algorithms function by generating decision trees, based on a specific set of the labelled input data. Furthermore, this decision tree undergoes a classification process, and hence, the C4.5 is generally called as the statistical classifier. Their results for the true positives and the false positives also revealed that the system showed an improvement in its performance. Furthermore, their approach suggested that it was scalable and required a lesser computation, which indicated that the classifiers were appropriately selected [30].

4 Proposed work and implementation

Here, the authors have discussed the proposed model, where they implemented the J48 algorithm for improving the anomaly detection in the IDS. Furthermore, they have also discussed the simulation environment and the experimental dataset used along with the feature selection technique and the 10-fold cross validation method that was applied.

4.1 NSL KDD dataset

The NSL KDD is a more advanced or refined version of the KDD CUP. It consisted of all the attributed needed to form the KDD. This NSL KDD is seen to be an open source programme and could be easily downloaded [31]. The major advantage of using the NSL KDD is that it did not contain a huge amount of redundant data and also consisted of an adequate number of records for training and testing the data. The NSL KDD data comprises of a complete training dataset with 125,973 records along with a complete testing dataset having 22,544 records [32]. Every record in the NSL KDD consisted of 42 attributes [33] which have been categorised as normal, binary and numeric, where the last attributes were also added as the class. 2 types of classes were present, i.e., Normal and Anomaly. The anomaly class is categorised into DOS, PROBE, R2L and U2R.

Some advantages of using the NSL-KDD instead of the original KDDCUP'99 dataset have been described below [33]:

- The redundant records have been removed from the training and the testing sets.
- The number of the selected records from every difficulty level was seen to be inversely proportional to their percentage value.
- The NSL-KDD comprised of a sufficient number of examples for the training and the testing sets and was therefore more affordable during experimentation.

The records contain 23 classes of different network attacks, i.e., normal and 22 forms of attacks: ftp write, guess passwd, neptune, imap, ipsweep, warezmaster, warezclient, teardrop, spy, portsweep, nmap, smurf, satan, pod, back, buffer overflow, phf, land, multihop, rootkit, perl and loadmodule [2].

4.2 Environment for knowledge analysis (WEKA)

WEKA is a widely used machine learning workbench that is coded in the Java language. It contains many machine learning algorithms, which were developed using Java, for carrying out many data mining processes and was developed by the Machine Learning Group at the University of Waikato in New Zealand [34]. The WEKA tool is open source

software which is available as per the GNU general public license (GPL) [35]. This is not one single programme but contains many algorithms and the GUI tool for carrying out data analysis and for predictive modelling. These algorithms can be directly applied to the dataset or could be modified using your Java code. WEKA comprises of many tools for the data mining activities like the classification, data pre-processing, clustering, regression, association rules, or visualisation. This tool also helps in developing many additional machine learning techniques. It is seen to contain an Experimenter, Explorer, Simple Command Line Interface, Knowledge flow, Java interface [36]. Furthermore, WEKA also contains many classes which could be easily accessed by other WEKA classes. The essential WEKA classes are the attribute and the instance. The attribute can be represented by any object of the class attributes that contains the attribute name, type and the nominal attributes values [37].

4.3 Feature selection

The feature selection procedure is generally used for improving the effectiveness of all the data mining algorithms and the performance of data classification [38]. The dataset contains numerous features, but not all of them are essential. Some features are redundant or irrelevant, where the redundant features do not provide any additional information while the irrelevant features do not provide any helpful information with regards to the context. The feature selection is based on a specific criterion used for choosing a subset of original features and employs techniques which are frequently applied in the data mining procedure for reducing dimensions [39].

Furthermore, feature selection process is also used for decreasing the number of the features by eliminating the redundant, irrelevant or the noise features. This is particularly useful as the irrelevant features increase the model complexities and the convergence time needed for a good model structure. The feature selection process also is seen to speed the learning or modelling process, improve the learning accuracy or quality and leads to better understanding of the model [40]. The process is categorised in 2 classes; filter and the wrapper approach [41]. The filter approach involves the selection of new feature subset which is dependent on the standard data characteristics. This approach ranks all the features based on specific statistical criteria. Thus, the features with the highest rank and high priority are selected; while the features with the least rank and a lower priority are not selected. This approach also helps in determining the classifier accuracy. The attribute evaluator and the ranker techniques are applied for ranking all the dataset features [40].

On the other hand, the wrapper approach develops all possible feature subsets based on the subset evaluator which uses the search methods. The performance of the classification method in the wrapper approach is also used as an evaluation criterion. The feature subset showing the best performance is selected using the various classification algorithms. For instance, if 10 features are available, the wrapper approach tries to find the subset having all the 10 features.

- 1st attribute: 3 features
- 2nd attribute: 3 features
- 3rd attribute: 4 features

Also, the classifier is used for all the subsets to determine which of the subsets shows the highest detection accuracy rate [40,42].

In this study, 3 feature selection processes have been used which have been selected based on the earlier studies. These processes were selected depending on the fact that they showed a reduction in the number of the features but still showed an effective and better IDS performance. These feature selection processes include information gain (IG) attribute evaluation which determines the IG for every attribute and evaluates its importance and relevance to the class label [43] the gain ratio (GR) attribute which used the gain extension information and split this data for evaluating the gain ratio. This split information splits the training data set, S, into v partitions which correspond to v outcomes for any test carried out on the attribute. The GR chooses attributes which show a higher splitting value [44]. Finally, the last selection process includes the correlation-based feature selection (CFS), which is used for identifying the similarity and the dependence between every feature and the linear relationship is determined which measures the dependence present between the features or the variables. This linear relationship shows a value between -1 to $+1$. When the value is 1, it indicates that the features are fully correlated; while the -1 value shows that the features were not correlated. A value of 0 shows that the features are completely independent [44].

4.4 The proposed approach

Here, in this study, the authors have proposed an enhanced J48 algorithm which has been developed based on the Decision Tree J48 algorithm for enhancing the intrusion detection accuracy and the IDS performance. One of the major problems in the construction of the decision tree involves the split value of the node, wherein the split value is the condition for dividing the data into 2-more subsets. The 1st split is known as the root node, while the rest of the splits are known as the leaf nodes (also called as the terminal or the decision nodes). Every internal is seen to split the space into 2-more subspaces depending on the discrete function for the input attribute values [45]. The split values provide an effective method for building the decision tree. For

obtaining a smaller and more effective decision tree, the split must be based on the maximal gain. This proposed algorithm introduced a novel approach for the selection of the split values, estimation of the IG and the GR for constructing the decision tree. Here, the attribute having the maximal normalised IG is utilised and the algorithm is seen to recur using small subsets. The split procedure stops when all the examples in the subset are seen to belong to one class. In the study, the authors have tried to use the standard deviation coefficients as a significant factor for improving the J48 algorithm.

The standard deviation (SD) can be defined as the number used for measuring how much the group is spread from the mean (average) value or from the expected value. The SD measures the data dispersement, which describes the spread of the data based on the mean value [46]. Furthermore, SD describes the class distribution. If the attribute has a low SD value, it shows that the data value is closer to the value of the mean and has a simple distribution; whereas if the attribute shows a high SD value, it shows that the data value is more widespread from the mean and is highly randomised. Furthermore, the Entropy and SD are different parameters, however, in many cases (not all) the Entropy is seen to be dependent on the SD of distribution [47]. The IG and the entropy values along with SD help in deciding on which attributes the data is split while constructing the tree.

In this study, the NSL KDD dataset is used, which consists of 41 features, all of which have a different importance while constructing the decision tree. Some features have a high SD value and they can affect the building of an effective decision tree as they have more widespread values. In this study, the authors observed that the features showing a low SD could help in building a more effective decision tree. Here, the authors have used SD as a significant coefficient along with information entropy and the split information for any attribute. This helped them select important and essential features which could affect the construction of the decision tree. Hence, if the attribute shows a large SD, the information entropy was multiplied with a large SD coefficient, while the split information was multiplied with a small SD coefficient. On the other hand, if any attribute showed a smaller SD value, the information entropy and split information were multiplied by the small SD coefficient. The following equations are used for estimating the information entropy and the split information:

$$\text{Information entropy} = \text{Entropy} = \sum_{i=0}^{x} \text{Std\_entrophy}^* - \text{pi} \log 2 \text{ pi} \quad (1)$$

$$\text{Information spilt} = \sum_{i=0}^{x} \text{Std\_spilt}^* - \text{pi} \log 2 \text{ pi} \quad (2)$$

Case 1: If standard deviation >500
Std_entropy=Math.log10 (stdDev)/10;
Std_split= Math.log10 (stdDev)/20;

Case 2: If standard deviation >200 and standard deviation <500
Std_entropy =stdDev/300;
Std_split =(stdDev)/300;

Case 3: If standard deviation <200
Std_entropy = (stdDev)/10;
Std_split = (stdDev)/10;

Fig. 2 Calculation of the SD coefficients

The process optimises the information entropy and the split node, while the J48 algorithm is used for building decision trees by selecting the attributes showing information entropy GR for the current split nodes. In this approach, the SD coefficients would be calculated for the numeric attributes in 3 cases as shown in Fig. 2.

The SD coefficient value is determined using statistical analysis. After carrying out >150 experiments, the best SD coefficient value was obtained as described in Fig. 2 .

The value showing the highest GR was selected as a split value for that specific node. Rather than carrying out multiple calculations, the authors used a very simple and effective approach rather than a more difficult and complicated approach. In their approach (Fig. 3), no need was seen to sort the various attribute values for estimating the split value. In Fig. 3, the authors have described the different steps used in their proposed algorithm.

4.4.1 Functionality overview of the proposed approach without using feature selection

The steps below were followed for developing a very effective IDS process using the improved J48 algorithm, which showed better performance and accuracy:

1. Selecting an appropriate dataset containing quality data like the NSL KDD. Section 4.1 presents more details about the NSL KDD dataset
2. Using a proper dataset for training and testing in the experiments.
3. Constructing an improved J48 algorithm model for building an effective classifier. The steps required for developing a classification system are as follows:
    a. The training classifier is seen as a learning step for building a classifier or a model.
    b. The classification of the test data helps in determining the accuracy of the classification rules. Also, if an acceptable accuracy is seen, these classification rules are used for new data.

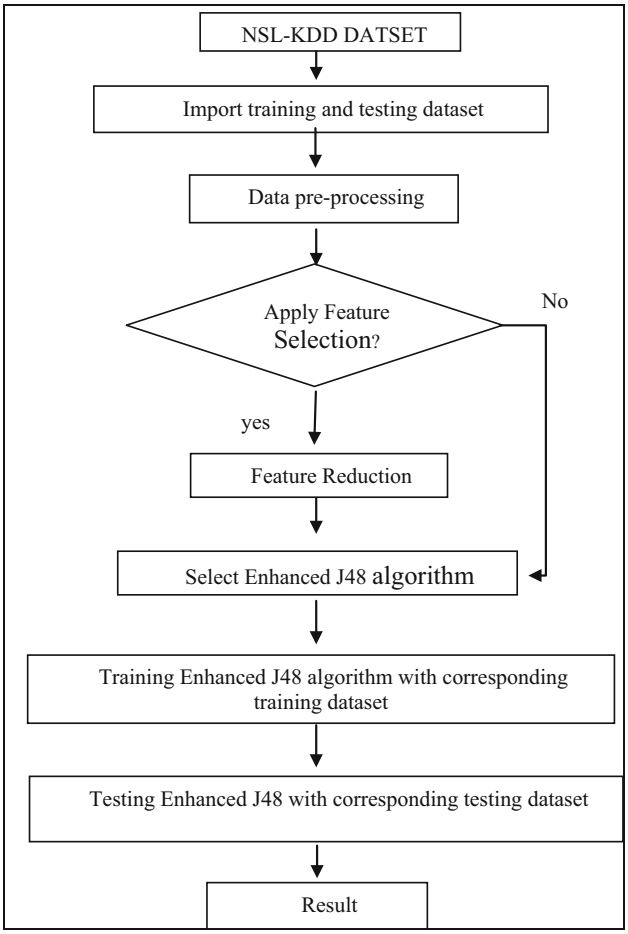

Fig. 3 An architecture of the proposed algorithm using the IDS approach

4. Generating a model which shows the maximal accuracy and the best performance.

Figures 4, 5 and 6 present the Algorithms 1, 2 and 3 which describe the pseudo code for the proposed approach without using the Feature selection.

4.4.2 Functionality overview of the proposed approach using feature selection

The steps described below outline the process for developing an efficient IDS process using the improved J48 algorithms which show a better performance and accuracy:

Selecting an appropriate dataset containing quality data like the NSL KDD. Section 4.1 presents more details about the NSL KDD dataset

Using a proper dataset for training and testing in the experiments.

Applying the Feature selection methods like the IG, GR and the CFS. Feature extraction involves the determination of the parameters that are used for providing a more precise character representation of a machine. This process helps in improving the accuracy and the performance of the classification process by selecting only the significant terms and deleting the noisy terms.

Constructing an improved J48 algorithm model for building an effective classifier. The steps required for developing a classification system are as follows:

The training classifier is seen as a learning step for building a classifier or a model.

The classification of the test data helps in determining the accuracy of the classification rules. Also, if acceptable accuracy is obtained, the classification rules are used for new data tuples

Generating a model which shows the maximal accuracy and the best performance.

Algorithm 1 Proposed Model using NSL_KDD Train

```
procedure model ()
InputFn= NSL-KDD data set possessing 41 features f1, f2, f3.....f42
 Use 41 features
 Use Enhanced J48
 Develop a robust model M
Propose the model
for every feature Fn
 Provide Fn to J48, NaiveBayes, ADTree, SVM, RandomTree, BayesNet and
DecisionStump using NSL-KDD Train+
      Calculate
       A1= J48 model accuracy
       A2= NaiveBayes model accuracy
       A3= ADTree model accuracy
       A4= SVM model accuracy
      A5= RandomTree model accuracy
      A6= BayesNet model accuracy
      A7= DecisionStump model accuracy
      E= Enhanced J48, J48, NaiveBayes, ADTree, SVM, RandomTree, REPTree and
      DecisionStump using NSL-KDD Train+

Compare of the accuracy of A1, A2, A3, A4, A5, A6, A7, E
Select the best model M= E
```

Fig. 4 Pseudo code of the proposed approach without using the feature selection

Algorithm 2 Proposed Model using NSL_KDD Test

```
procedure model ()
InputFn= NSL-KDD data set possessing 41 features f1, f2, f3.....f42
Use 41 features
Use Enhanced J48
Develop a robust model M
Propose the model
for every feature Fn
Provide Fn to J48, NaiveBayes, ADTree, SVM, RandomTree, BayesNet,
RandomForest, SimpleCart, ANN, NBTree and DecisionStump using NSL-
KDD Test+
Calculate
A1= J48 model accuracy
A2= NaiveBayes model accuracy
A3= ADTree model accuracy
A4= SVM model accuracy
A5= RandomTree model accuracy
A6= BayesNet model accuracy
A7= RandomForest model accuracy
A8= SimpleCart model accuracy
A9= ANN model accuracy
A10= NaiveBaye model accuracy
A11= NBTree model accuracy
A12= DecisionStump model accuracy
E= Enhanced J48, J48, NaiveBayes, ADTree, SVM, RandomTree, BayesNet,
      RandomForest, SimpleCart, ANN, NBTree and DecisionStump using
      NSL-KDD Test+
Compare of the accuracy of A1, A2, A3, A4, A5, A6, A7, A8, A9, , A10, A11,
A12, E
Select the best model M= E
```

Fig. 5 Pseudo code of the proposed approach without using the feature selection

Algorithm 3 Proposed Model using NSL_KDD Test 21

```
procedure model ()
InputFn= NSL-KDD data set possessing 41 features f1, f2, f3.....f42
Use 41 features
Use Enhanced J48
Develop a robust model M
Propose the model
for every feature F_n
Provide F_n to J48, NaiveBayes, ADTree, SVM, RandomTree, BayesNet,
RandomForest, SimpleCart, ANN, NBTree and DecisionStump using NSL-
KDD Test-21
Calculate
A1= J48 model accuracy
A2= NaiveBayes model accuracy
A3= ADTree model accuracy
A4= SVM model accuracy
A5= RandomTree model accuracy
A6= BayesNet model accuracy
A7= RandomForest model accuracy
A8= SimpleCart model accuracy
A9= ANN model accuracy
A10= NaiveBaye model accuracy
A11= NBTree model accuracy
A12= DecisionStump model accuracy
E= Enhanced J48, J48, NaiveBayes, ADTree, SVM, RandomTree, BayesNet,
RandomForest, SimpleCart, ANN, NBTree and DecisionStump using
NSL-KDD Test-21
Compare of the accuracy of A1, A2, A3, A4, A5, A6, A7, , A8, A9, , A10,
A11, A12, E
Select the best model M= E
```

Fig. 6 Pseudo code of the proposed approach without using the feature selection

Algorithm 4 Proposed Model using NSL_KDD Train

```
procedure model ()
InputFn= NSL-KDD data set possessing 41 features f1, f2, f3.....f42
Reduce 41 features to # of features based on a number of the proposed filters
Use Enhanced J48
Develop a robust model M
Propose the model
for every feature F_n
Provide F_n to J48, using NSL-KDD Train+
Calculate
A1= J48 model accuracy
E= Enhanced J48 and J48 algorithm using NSL-KDD Train+
Compare of the accuracy of A1and E
Select the best model M= E
```

Fig. 7 Pseudo code of the proposed approach using the feature selection

Figure 7 describes the Algorithm 4 which presents the pseudo-code for the proposed approach using the Feature selection

4.5 Implementation

For implementing the proposed approach, the authors used the Java language and the WEKA tool. They implemented the improved J48 algorithm on the computer with a 64-bit Windows 7 OS, 4 GB RAM and an i5 Intel core, and investigated the enhanced J48 algorithm. WEKA tool (described in Sect. 5.2) and the NetBeans [48] were used for carrying out all experiments. The model was conducted in the Weka environment, and hence, the authors extracted the WEKA Jar file from the WEKA file, as this file consisted of all the Weka classifiers and the clustering algorithms. The J48 algorithm comprises of 23 classes and this algorithm file was selected and imported in the NetBeans for enabling the modifications to be made in the information entropy, GR and the split information for improving the construction of the decision tree and selection of the split node. 4 classes were used for making improvements (i.e., C45Split.java, GainRatioSplitCrit.java, EntropyBasedSplitCrit.java, and the InfoGainSplitCrit.java). The C45Split.java was used for implementing a C4.5-type split in the attribute, while GainRatioSplitCrit.java was used for estimating the GR for a particular distribution. The EntropyBasedSplitCrit.java was an Abstract class which was used for computing the splitting criteria depending on the entropy of the class distribution and the InfoGainSplitCrit.java file was used for estimating the IG for a specific distribution. The authors also used the Ant external libraries [49] for creating the Jar file for the WEKA environment as it would help in utilising the modifications that were made in the J48 algorithm.

The authors carried out the experiments for comparing the performance of the proposed J48 algorithm and various tree-based classifiers. They carried out the analysis using various parameters like the time need by the classifier for model construction, true positive rate, the false positive rate, and the accuracy. The true positive (TP) is defined as the examples used in the study that were correctly predicted to be normal. The true negatives (TNs) represent the examples which were correctly predicted to be an attack. The false positives (FPs) are defined as the examples which were falsely predicted to be an attack when they were normal; whereas the false negatives (FNs) are defined as examples which were presumed to be normal, but they were actually an attack. The accuracy of the system is described as the number of the correct predictions made by the system. It is computed as follows:

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

5 Experimental analysis and results

WEKA utilises many general data mining techniques such as data cleaning, classification, clustering, data pre-processing, regression, visualisation, and feature selection. As stated earlier WEKA us a type of automated data mining technique used for carrying out many classification experiments using the NSL-KDD dataset. In this section, the authors have analysed all the experimental results obtained. They used the WEKA tool for evaluating the proposed algorithm and compared their results with those obtained using standard algorithms using similar values and parameters. This comparison was carried out using parameters like the accuracy of the technique in detecting the attacks which affected the training and the testing NSL KDD dataset. Furthermore, the authors also tested various different classifiers like Naïve Bayes, CART, RandomForest, NBTree, AD Tree, and SVM in addition to their proposed algorithm using the NSL KDD

dataset. Furthermore, they also determined the accuracy of the predictive models using the 10-fold cross-validation and the supplied test set. In the 10-fold cross validation method, the authors randomly divided their samples into 10 different subsamples, out of which 4 of the samples were used for testing purposes, while the rest of the 6 were used for the training purposes. Furthermore, the supplied classifier test set was also investigated on its performance regarding how it predicted the class of an example set which was loaded from the file. This was done because each of the subsamples was used for testing and training. Also, the authors compared their results to all the published studies between 2013–2017 with and without the help of the feature selection techniques. Also, they determined the accuracy and the robustness of their approach using various parameters. Their results proved that their proposed enhanced J48 algorithm showed a better accuracy and performance in comparison to the other techniques. They noted 100% accuracy for the 10-fold cross validation method, and a 90% accuracy using the supplied tests as shown below. Also, the enhanced and improved J48 algorithm showed a better performance than the individual performances of J48 (C4.5) and such other algorithms.

For obtaining an accurate detection, several experiments were carried out using the proposed and many other approaches. Continuous experiments were carried out till stable results and a better accuracy was obtained. In this section, the authors have discussed the experimental setup and their results analysis.

5.1 10-fold cross validation test

In this study, the WEKA tool was used for investigating the enhanced J48 algorithm and the results were compared to those obtained using a standard algorithm under similar experimental conditions. Here, the authors have compared their results to those obtained using many algorithms like the J48, NaiveBayes, SVM, ADTree, BayesNet, RandomTree and DecisionStump using a similar experimental setup. Table 1 compares the results for the intrusion detection accuracy for all the studied algorithms. This experiment also used the full training NSL KDD (KDDTrain+) dataset having 41 features and the 10-fold cross validation test.

As seen in Table 1, the enhanced J48 algorithm showed the highest detection accuracy of 99.88% for all the 41 features; while BayesNet showed the lowest accuracy of 74.432%.

In Table 2, the authors have compared the intrusion detection accuracy of the enhanced J48 algorithm with the ensemble model proposed earlier Shrivas and Mishra [50].

In their study, Shrivas and Mishra [50] proposed an ensemble model which comprised of the C4.5 (J48) and the CART model. In this study, the 10-fold cross-validation test has been used for testing this model. A part of the dataset was also used for testing purposes, whereas the remaining part of the data was used for training the proposed ensemble model of C4.5 and CART, where 20% of the NSL-KDD dataset was used. Furthermore, in their study, Shrivas and Mishra [50] classified the NSL-KDD dataset in 2 different sections, i.e., NSL-KDD with a binary and a multiclass.

Here, the authors have compared the C4.5 (J48) and the CART model ensemble proposed by Shrivas and Mishra [50] with the enhanced J48 algorithm and a CART model algorithm using similar experimental parameters and the classified NSL-KDD data set and the results are presented in Fig. 8.

Table 1 Accuracy analyses results carried out by a 10-fold cross-validation test for different models

Detection accuracy	
Naïve bayes	90.3829
J48	99.78
ADTree	98.4902
SVM	97.405
RandomTree	99.7658
BayesNet	74.4322
Enhanced J48 algorithm	99.88
DecisionStump	92.215

Table 2 Accuracy analyses of the comparison between the enhanced J48 algorithm and the ensemble model proposed earlier [50]

Detection accuracy		
Model	Binary class (%)	Multiclass (%)
C4.5 [50]	99.56	99.46
CART [50]	99.66	99.46
C4.5+CART [50]	99.67	99.53
Enhanced J48 algorithm	99.62	99.38
Enhanced algorithm +CART	99.71	99.65

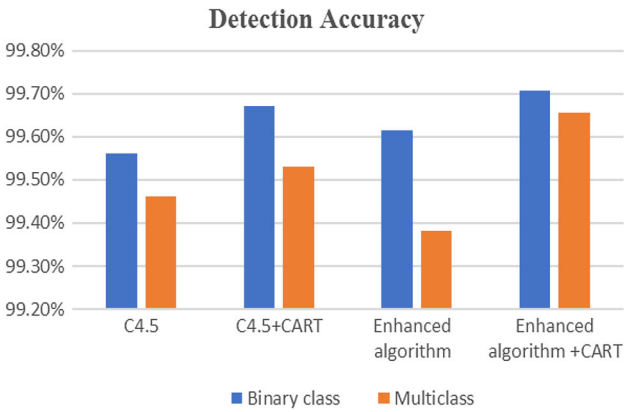

Fig. 8 A comparison between the accuracy analyses between the enhanced J48 algorithm and the ensemble model proposed earlier [50]

After comparing all the results as shown in Fig. 8, the authors observed that the combination of the enhanced J48 algorithm along with the CART model showed a higher accuracy as compared to the ensemble model of C4.5 (J48) and the CART proposed earlier [50]. The combination of the Enhanced J48 algorithm and the CART model showed a 99.71% accuracy for all the 41 features in the Binary class and a detection accuracy of 99.65% for the Multiclass. These values were higher than those shown by the ensemble model proposed earlier [50], which showed a 99.67% detection accuracy using all the 41 features for the Binary class and also a 99.53% accuracy for the Multiclass. The enhanced J48 algorithm used in this study showed a better performance and intrusion detection than the ensemble model proposed earlier which used the C4.5 (J48) and the CART model [50].

5.2 Supply test without using the feature selection technique

Here, the authors have compared the proposed enhanced J48 algorithm with other algorithms like the J48, NaiveBayes, ANN, SVM, ADTree, BayesNet, RandomTree, DecisionStump, RandomForest, SimpleCart, and the NBTree under similar values and parameters with the help of the supply test set.

Table 3 shows the comparison results for the detection accuracy for all the algorithms used in the IDS. The authors used a Full Train NSL KDD dataset (KDDTrain+.) along with the Full Test NSL KDD (KDDTest+.) dataset having 41 features and a supplied test set.

As seen from the Table 3, the enhanced J48 algorithm showed the highest detection accuracy of 90.01% for all the 41 features; while the least accuracy was shown by ADTree (74.432%). After comparing the enhanced J48 and the J48 algorithm, the enhanced J48 algorithm showed better values as compared to the J48 algorithm (81.5%). Hence, out of the studied 11 algorithms, it was seen that the enhanced J48 algorithm showed the best accuracy for the IDS.

Table 3 Accuracy analyses of various models using the supplied test and full test NSL KDD dataset

Detection accuracy	
NaiveBayes	76.1178
J48	81.5339
ADTree	74.308
SVM	75.3948
RandomTree	81.3565
BayesNet	74.4322
DecisionStump	79.9858
RandomForest	80.1899
SimpleCart	80.3229
ANN	79.3559
NBTree	79.6842
Enhanced J48 algorithm	90.02

Table 4 Accuracy analyses of various models using the supplied test and Test-21 NSL KDD dataset

Detection accuracy	
NaiveBayes	54.8861
J48	64.903
ADTree	51.308
SVM	53.3249
RandomTree	64.7764
BayesNet	51.4599
DecisionStump	63.1983
RandomForest	62.8101
SimpleCart	62.6245
ANN	57.6034
NBTree	64.2278
Enhanced J48 algorithm	76.2363

In Table 4, the authors have also compared the intrusion detection accuracy of the proposed J48 algorithm with many other data mining techniques. Here, the authors have compared the proposed enhanced J48 algorithm with other algorithms like the J48, NaiveBayes, ANN, SVM, ADTree, BayesNet, RandomTree, DecisionStump, RandomForest, SimpleCart, and the NBTree under similar values and parameters. They further used a Full Train NSL KDD dataset (KDDTrain+) along with a Test-21 NSL KDD (KDDTest-21) dataset with 41 features and a supplied test set for the experiment.

As seen from the Table 4, the enhanced J48 algorithm showed the highest detection accuracy of 76.2363% for all the 41 features; while the least accuracy was shown by BayesNet (51.4599%). After comparing the enhanced J48 and the J48 algorithm, the enhanced J48 algorithm showed better values as compared to the J48 algorithm (64.9%). All the other algorithms, except the proposed J48 algorithm, showed a detection accuracy ranging between 50-65%. Hence, out of the studied 11 algorithms, the proposed J48 algorithm showed the best accuracy for the IDS.

Furthermore, the authors also compared the performance of the enhanced J48 algorithm with the algorithm proposed earlier [44]. The comparison results for the intrusion detection accuracy rate between the proposed J48 algorithm and the proposed algorithm [44], carried out using similar values and parameters, have been presented in Table 5. They used a Full train (KDDTrain+), full test (KDDTest+) and the test-21(KDDTest-21) datasets in the experiment, along with the supplied test set.

Table 5 Comparison of the detection accuracy for the enhanced J48 algorithm and the approach proposed earlier [44]

Algorithm type	All feature using full test	All feature using test-21
J48 Algorithm [44]	81.05	63.654
Naïve bayes [44]	76.65	55.77
NB-tree [44]	82.02	66.16
Multi-layer perception [44]	77.41	57.34
SVM [44]	69.52	42.29
SimpleCart [44]	80.3229	62.6245
Enhanced J48 algorithm	90.02	76.2363

In their study, Bajaj and Arora [44] suggested applying the feature selection technique for decreasing the NSL KDD dataset dimensions using the feature reduction and the machine learning approach. They used many discriminative machine learning algorithms like the J48, SVM, NaïveBayes, Multilayer Perception, NBTree, and the SimpleCart. In this study, the authors have compared the detection accuracy of the enhanced J48 algorithm with the approach proposed by Bajaj and Arora [44] without using the feature selection technique for the NSL-KDD data set having 41 features along with a single class of Labels and the results are presented in Fig. 9.

In their study, Bajaj and Arora [44] observed the maximum value for detection accuracy of 82.3225%, shown by SimpleCart algorithm for the full test dataset, while it showed an accuracy of 66.7764% for the test-21 dataset after the features were decreased to 33 from the original 41 and using a single class of labels.

After comparing the enhanced J48 algorithm with those described by Bajaj and Arora [44], the authors observed that their enhanced J48 algorithm showed a maximal accuracy of 90.02% for the Full Test and 76.2363% for the test-21 dataset using the 41 features. Hence, the enhanced J48 proved

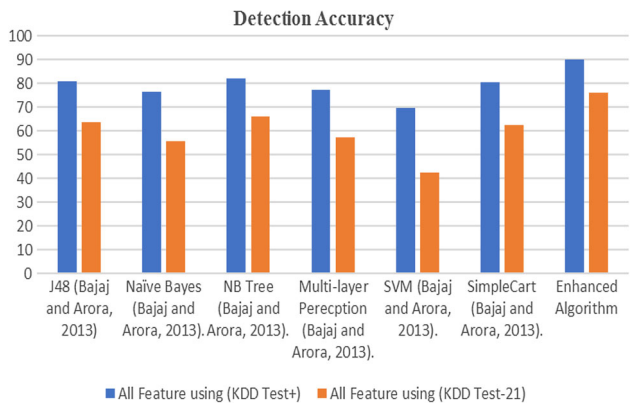

Fig. 9 Comparison of the detection accuracy for the enhanced J48 algorithm and the approach proposed earlier [44]

Table 6 Comparison of the detection accuracy for the enhanced J48 algorithm and the approach proposed earlier [51]

Detection accuracy	
Hoeffding Tree [51]	79.0454
J48 algorithm [51]	74.7028
RandomForest [51]	77.8921
RandomTree [51]	74.2814
REPTree [51]	75.3504
Enhanced J48 algorithm	90.02

to be the better algorithm showing a higher detection accuracy in the IDS when compared to the approach described earlier [44].

Table 6 presents the comparison results for the detection accuracy when the enhanced J48 algorithm was compared to the other tree-based data mining algorithms proposed by Elekar and Waghmare [51]. The authors have used a Full train (KDDTrain+) and a test (KDDTest+) data set for the comparison along with the supplied test set.

In their study, Elekar and Waghmare [51] had proposed a comparison between many tree-based data mining algorithms like the J48, Hoeffding Tree, RandomTree, RandomForest and the REPTree. They concluded that the Hoeffding Tree algorithm was the best tree-based algorithm for the IDS with an accuracy of 79.0454 %.

Here, we have compared the Enhanced J48 algorithm to those studied earlier [51]. Table 6 presents the comparison results for the studies and it can be seen that the enhanced J48 algorithm showed the best detection accuracy of 90.02%, which was higher than that seen for the Hoeffding Tree (79.0454%) and other mining algorithms proposed earlier [51]. Hence, the enhanced J48 algorithm is the best-suited algorithm in the IDS with the maximal accuracy.

In another experiment, we have also compared the enhanced J48 algorithm to the algorithm proposed earlier [12] using similar values and parameters. They used the Full train (KDDTrain+), the Full test (KDDTest) and the Test-21(KDDTest-21) datasets along with the supplied test set. In Table 7, the comparison results for the intrusion detection accuracy rate between the proposed J48 algorithm and the approach described by Ashfaq et al. [12] have been described.

In their study, Ashfaq et al. [12] developed a novel fuzziness-based semi-supervised learning approach using unlabelled samples and a supervised learning algorithm for improving the classifier performance in the IDS, and they observed a high detection accuracy of 82.41% for their algorithm.

We have compared the detection accuracy of the enhanced J48 algorithm with that proposed by Ashfaq et al. [12]. As described in Fig. 10, the proposed J48 algorithm showed the maximal detection accuracy of 90.02%, which was bet-

Table 7 Comparison of the detection accuracy for the enhanced J48 algorithm and the approach proposed earlier [12]

Detection accuracy	KDDTest	KDDTest-21
J48 algorithm [12]	81.05	63.97
Naive Bayes algorithm [12]	76.56	55.77
NB tree algorithm [12]	82.02	66.16
RandomForest algorithm [12]	80.02	63.25
RandomTree algorithm [12]	81.59	58.51
Multi-layer perceptron [12]	77.41	57.34
SVM [12]	69.52	42.29
Proposed fuzziness algorithm [12]	82.41	67.82
Enhanced J48 algorithm	90.02	76.2363

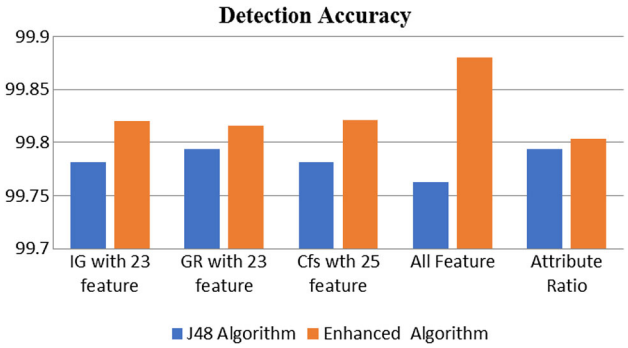

Fig. 11 A comparison between the performances of the J48 and the enhanced J48 algorithm

ter than the fuzziness-based semi-supervised learning [12] and the other classifiers used. Hence, after comparing the enhanced J48 algorithm with the proposed approach, it was seen to be more accurate and reliable for classifying the anomaly or intrusion detection within the IDS.

As shown in all the above experiments, the enhanced J48 algorithm showed a must better and efficient performance as compared to the other data mining approaches. The algorithm was compared to the approaches proposed earlier using many tests and various test data sets and it still showed a better performance and showed higher intrusion detection accuracy. Also, all the results indicated that the proposed J48 algorithm was reliable and more suited to be used as the main basis for the classification of anomaly detection in any IDS.

After comparing the experimental result for the enhanced J48 algorithm with the feature selection technique proposed by Chae et al. [52] and presenting the results in Fig. 11, it can be seen that the enhanced J48 algorithm showed higher and accurate detection for the KDDTrain+ dataset and the 10-fold cross validation test than those reported earlier [52]. Hence, it can be noted that the Enhanced J48 algorithm is the best algorithm within any IDS as it showed the maximal intrusion detection accuracy rate.

6 Conclusions and future work

Finally, the conclusions of the study and the summary of the results obtained from all the experiments carried out using the enhanced J48 algorithm for anomaly detection have been presented in this section. Also, we have described some further work that needs to be carried out.

In today's day and age, the prevention of the security breaches with the help of currently available technology is quite unrealistic. Hence, intrusion detection is a very important feature in the network security. Furthermore, the misuse detection methods are unable to detect the unknown attacks; hence, anomaly detection needs to be used for identifying such attacks. The data mining technique is applied in the

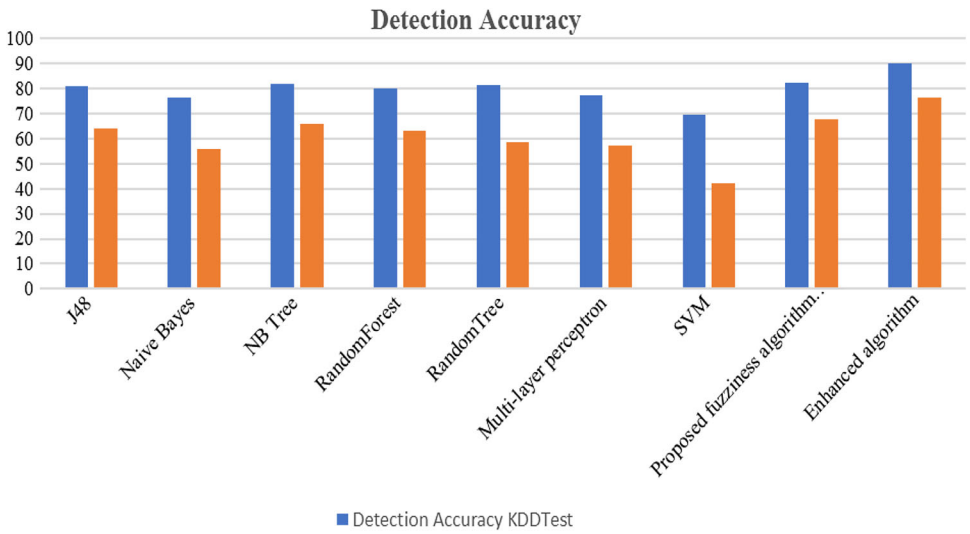

Fig. 10 Comparison of the detection accuracy of the enhanced J48 algorithm and the proposed approach in [12]

anomaly-based detection techniques for improving the intrusion detection accuracy rate.

In this article, we have developed and proposed the Enhanced J48 Classification Algorithm for the intrusion and anomaly detection. We also have compared the results of this algorithm with many other data mining approaches and it was seen that the proposed algorithm showed a better performance. This new method was very effective for detection of many attacks and showed higher detection accuracy, as compared to the algorithms reported earlier.

This proposed technique could classify the data as either normal or abnormal. It showed a detection accuracy of 99.88% for the 10-fold cross validation test when using the full train dataset, an accuracy of 90.02% for the supplied test set when applying the full train and test dataset and 76.23% accuracy when using the full train and the test-21 dataset, and it was higher than the other reported techniques. The detection accuracy of the enhanced J48 algorithm was also improved with the help of the SD coefficient, which further improved the gain ratio, entropy and the election of the split value. This split value divided the data into 2-more subsets. Thus, based on the results, it was concluded that the proposed enhanced J48 approach is very simple and effective in decreasing the false alarm ratio and improving the intrusion detection accuracy.

Finally, for future work, the IDS intrusion detection accuracy rate and the performance of the proposed technique have to be improved and it has to be implemented in real network environments. The authors would also like to further explore the features of the J48 algorithm and improve the split value and the construction of the decision tree by applying the correlation feature selection technique.

References

1. Agrawal, S., Agrawal, J.: Survey on anomaly detection using data mining techniques. Procedia Comput. Sci. **60**, 708–713 (2015)
2. Sheta, A.F., Alamleh, A.: A Professional Comparison of C4.5, MLP, SVM for Network Intrusion Detection Based Feature Selection Analysis (2015)
3. Onik, A.R., Haq, N.F., Alam, L., Mamun, T.I.: An analytical comparison on filter feature extraction method in data mining using J48 classifier. Int. J. Comput. Appl. **124**(13) (2015)
4. Kumar, G.R., Nimmala, M., Narasimha, G.: An approach for intrusion detection using novel Gaussian based kernel function. J. Univers. Comput. Sci. **22**(4), 589–604 (2016)
5. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann (2016)
6. Panda, M., Patra, M.R.: Network intrusion detection using Naïve bayes. Int. J. Comput. Sci. Netw. Secur. **7**(12), 258–263 (2007)
7. Weiming, H., Wei, H., Maybank, S.: AdaBoost-based algorithm for network intrusion detection. IEEE Trans. Syst. Man Cybern. B Cybern. **38**, 577–583 (2008)
8. Kosamkar, V.: Improved Intrusion detection system using C4.5 decision tree and support vector machine. Doctoral dissertation, Mumbai University (2013)
9. Li, W., Yi, P., Wu, Y., Pan, L., Li, J.: A new intrusion detection system based on KNN classification algorithm in wireless sensor network. J. Electr. Comput. Eng. 1–7 (2014). doi:10.1155/2014/240217
10. Ahmed, M., Mahmood, A.N., Hu, J.: A survey of network anomaly detection techniques. J. Netw. Comput. Appl. **60**, 19–31 (2016)
11. Pathan, A.S.K. (ed.).: The State of the Art in Intrusion Prevention and Detection. CRC Press (2014)
12. Ashfaq, R.A.R., Wang, X.Z., Huang, J.Z., Abbas, H., He, Y.L.: Fuzziness based semi-supervised learning approach for intrusion detection system. Inf. Sci. **378**, 484–497 (2017)
13. Breiman, L., Friedman, J., Stone C., Olshen, R.: Classification and Regression Trees. The Wadsworth and Brooks-Cole Statistics-Probability Series. Taylor and Francis (1984)
14. Quinlan, J.R.: C4. 5: Programs for Machine Learning. Elsevier (2014)
15. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques, 3rd edn. Morgan Kaufmann Publishers Inc., San Francisco (2012)
16. Ooi, S.Y., Leong, Y.M., Lim, M.F., Tiew, H.K., Pang, Y.H.: Network intrusion data analysis via consistency subset evaluator with ID3, C4.5 and bestfirst trees. IJCSNS **13**(2), 7 (2013)
17. Medhat, K., Ramadan, R.A., Talkhan, I.: Security in mission critical communication systems: approach for intrusion detection. In: Multimedia Services and Applications in Mission Critical Communication Systems, pp. 270–291. IGI Global (2017)
18. Sahu, S., Mehtre, B.M.: Network intrusion detection system using J48 decision tree. In: 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 2023–2026. IEEE (2015, August)
19. Panda, M., Abraham, A., Patra, M.R.: A hybrid intelligent approach for network intrusion detection. Procedia Eng. **30**, 1–9 (2012)
20. Aburomman, A., Reaz, M.: A novel SVM-kNNPSO ensemble method for intrusion detection system. Appl. Soft Comput. J. **38**, 360–372 (2016)
21. Goeschel, K.: Reducing false positives in intrusion detection systems using data-mining techniques utilizing support vector machines, decision trees, and naive Bayes for off-line analysis. SoutheastCon 2016, Norfolk, pp. 1–6 (2016)
22. Sharma, S., Gupta, A., Agrawel, S.: A survey of intrusion detection system for denial of service attack in cloud. Int. J. Comput. Appl. **19**, 1–4 (2015)
23. Al Kaabi, S., Al Kindi, N., Al Fazari, S., Trabelsi, Z.: Virtualization based ethical educational platform for hands-on lab activities on DoS attacks. 2016 IEEE Global Engineering Education Conference (EDUCON), pp. 273–280 (2016)
24. Noureldien, N., Yousif, I.: Accuracy of machine learning algorithms in detecting DoS attacks types. Sci. Technol. **6**(4), 89–92 (2016)
25. AbdJalil, K., Mara, S.: Comparison of machine learning algorithms performance in detecting network intrusion. In: Proceedings of Networking and Information Technology (ICNIT), pp. 221–226. Manila (2010)
26. Jain, Y.K., Upendra: An efficient intrusion detection based on decision tree classifier using feature reduction. Int. J. Sci. Res. Publ. **2**(1), January (2012)
27. Mazraeh, S., Modhej, A., Neysi, S.H.N.: Intrusion detection in computer networks using combination of machine learning techniques. Int. J. Comput. Sci. Netw. Secur. (IJCSNS) **16**(8), 122 (2016)
28. Gaikwad, D.P., Thool, R.C.: Intrusion detection system using bagging ensemble method of machine learning. In: 2015 International

Conference on Computing Communication Control and Automation (ICCUBEA), pp. 291–295. IEEE (2015, February)
29. Nema, A., Tiwari, B., Tiwari, V.: Improving accuracy for intrusion detection through layered approach using support vector machine with feature reduction. In: Proceedings of the ACM Symposium on Women in Research 2016, pp. 26–31. ACM (2016, March)
30. Modi, U., Jain, A.: An improved method to detect intrusion using machine learning algorithms. Inf. Eng. Int. J. **4.2**, 17–29 (2016)
31. [Online]. Available: https://www.unb.ca/research/iscx/dataset/iscx-NSL-KDD-dataset.html [Accessed 26 April 2017]
32. Chaudhari, R.R., Patil, S.P.: Intrusion Detection System: Classification, Techniques and Datasets to Implement (2017)
33. Aljawarneh, S., Aldwairi, M., Yasin, M.B.: Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. J. Comput. Sci. (2017)
34. Smith, T.C., Frank, E.: Introducing machine learning concepts with WEKA. Stat. Genom. Methods Protoc. 353–378 (2016)
35. [Online]. Available Weka: http://www.cs.waikato.ac.nz/ml/index.html. [Accessed 26 April 2017]
36. Alcala-Fdez, J., Garcia, S., Fernandez, A., Luengo, J., Gonzalez, S., Saez, J. A., Triguero, I., Moyano, J.M., Jesus, M.J., Sanchez, L., Herrera, F.: Comparison of KEEL versus open source Data Mining tools: Knime and Weka software (2016)
37. Bouckaert, R.R., Frank, E., Hall, M.A., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: WEKAâ˜ experiences with a java open-source project. J. Mach. Learn. Res. **11**(Sep), 2533–2541 (2010)
38. Ravage, U., Marathe, N., Padiya, P.: Feature selection based hybrid anomaly intrusion detection system using K means and RBF kernel function. Procedia Comput. Sci. **45**, 428–435 (2015)
39. De la Hoz, E., De La Hoz, E., Ortiz, A., Ortega, J., Prieto, B.: PCA filtering and probabilistic SOM for network intrusion detection. Neurocomputing **164**, 71–81 (2015)
40. Najafabadi, M.M., Khoshgoftaar, T.M., Seliya, N.: Evaluating feature selection methods for network intrusion detection with kyoto data. Int. J. Reliab. Qual. Saf. Eng. **23**(01), 1650001 (2016)
41. Xue, B., Zhang, M., Browne, W.N., Yao, X.: A survey on evolutionary computation approaches to feature selection. IEEE Trans. Evol. Comput. **20**(4), 606–626 (2016)
42. Eid, H.F., Hassanien, A.E., Kim, T.H., Banerjee, S.: Linear correlation-based feature selection for network intrusion detection model. In: Advances in Security of Information and Communication Networks, pp. 240–248. Springer, Berlin (2013)
43. Alhaj, T.A., Siraj, M.M., Zainal, A., Elshoush, H.T., Elhaj, F.: Feature selection using information gain for improved structural-based alert correlation. PLoS ONE **11**(11), e0166017 (2016)
44. Bajaj, K., Arora, A.: Improving the intrusion detection using discriminative machine learning approach and improve the time complexity by data mining feature selection methods. Int. J. Comput. Appl. **76**(1) (2013)
45. Oreski, D., Oreski, S., Klicek, B.: Effects of dataset characteristics on the performance of feature selection techniques. Appl. Soft Comput. **52**, 109–119 (2017)
46. Brown, G.W.: Standard deviation, standard error: which'standard'should we use? Am. J. Dis. Child. **136**(10), 937–941 (1982)
47. [Online]. Available https://math.stackexchange.com/questions/651077/is-standard-deviation-the-same-as-entropy. [Accessed 26 April 2017]
48. [Online]. Available: https://netbeans.org/ [Accessed 26 April 2016]
49. [Online]. Available: https://www.tutorialspoint.com/ant/ant_creating_jar_files.htm [Accessed 26 April 2016]
50. Shrivas, A.K., Mishra, P.K.: Intrusion detection system for classification of attacks with cross validation. Probe **2**(209), U2R (2016)
51. Elekar, K.S., Waghmare, M.M.: Comparison of tree base data mining algorithms for network intrusion detection. Int. J. Eng. Educ. Technol. **3**(2) (2015)
52. Chae, H.S., Jo, B.O., Choi, S.H., Park, T.K.: Feature selection for intrusion detection using NSL-KDD. Recent Adv. Comput. Sci. 184–187 (2013)

Shadi Aljawarneh is an associate professor, Software Engineering, at the Jordan University of Science and Technology, Jordan. He holds a B.Sc. degree in Computer Science from Jordan Yarmouk University, a M.Sc. degree in Information Technology from Western Sydney University and a Ph.D. in Software Engineering from Northumbria University-England. He worked as an associate professor in faculty of IT in Isra University, Jordan since 2008. His research is centered in software engineering, web and network security, e-learning, bioinformatics, Cloud Computing and ICT fields. Aljawarneh has presented at and been on the organizing committees for a number of international conferences and is a board member of the International Community for ACM, Jordan ACM Chapter, ACS, and IEEE. A number of his papers have been selected as "Best Papers" in conferences and journals.

Muneer Bani Yassein is an associate professor at the Department of Computer science at Jordan University of Science and Technology (JUST), He received his Ph.D. degrees in Computer Science from the University of Glasgow, U.K., in 2007 and M.Sc. in Computer Science, from Al Albayt, University, Jordan in 2001 and his B.Sc in Computing Science and Mathematics from Yarmouk University, Jordan in 1985. Bani Yassein served as chairman of the department of computer science from 2008 to 2010, as Vice Dean of the Faculty of Computer and Information Technology from 2010 to 2012, and from 2013–2014. Bani Yassein is currently conducting research in Mobile Ad hoc Networks, Wireless sensors Networks, Cloud Computing, Simulation and Modelling, Bani Yassein, has published over 60 technical papers in well reputed international journals and conferences. During his career, he has supervised more than 50 graduate and undergraduate students. Bani Yassein is a member of the technical programs of several international journals and conferences. In academic year 2014/15 he is spending his sabbatical year at Edinburgh Napier university, Edinburgh, U.K.

Mohammed Aljundi is a post graduate student at Jordan University of Science and Technology, Jordan. His research interests are IDS, Security and Performance.