Network Intrusion detection using Oversampling technique and Machine learning algorithms

Hafiza Anisa Ahmed [Corresp., 1] , **Anum Hameed** [1] , **Narmeen Zakaria Bawany** [1]

[1] Department of Computer Science and Software Engineering, Jinnah University for Women, Karachi, Sindh, Pakistan

Corresponding Author: Hafiza Anisa Ahmed
Email address: hafizaanisaahmed@gmail.com

The expeditious growth of the World Wide Web and the rampant flow of network traffic have resulted in a continuous increase of network security threats. Cyber attackers seek to exploit vulnerabilities in network architecture to steal valuable information or disrupt computer resources. Network Intrusion Detection System (NIDS) is used to effectively detect various attacks, thus providing timely protection to network resources from these attacks. To implement NIDS, stream of supervised and unsupervised machine learning approaches is applied to detect irregularities in network traffic and to address network security issues. Such NIDSs are trained using various datasets that include attack traces. However, due to the advancement in modern-day attacks, these systems are unable to detect the emerging threats. Therefore, NIDS needs to be trained and developed with a modern comprehensive dataset which contains contemporary common and attack activities. This paper presents a framework in which different machine learning classification schemes are employed to detect various types of network attack categories. Five machine learning algorithms: Random Forest, Decision Tree, Logistic Regression, K-Nearest Neighbors and Artificial Neural Networks, are used for attack detection. This study uses a dataset published by the University of New South Wales (UNSW-NB15), a relatively new dataset that contains a large amount of network traffic data with nine categories of network attacks. The results show that the classification models achieved the highest accuracy of 89.29% by applying the Random Forest algorithm. Further improvement in the accuracy of classification models is observed when Synthetic Minority Oversampling Technique (SMOTE) is applied to address the class imbalance problem. After applying the SMOTE, the Random Forest classifier showed an accuracy of 95.1% with 24 selected features from the Principal Component Analysis method.

Network Intrusion detection using Oversampling technique and Machine learning algorithms

6 Hafiza Anisa Ahmed[1], Anum Hameed[1], Narmeen Zakaria Bawany[1]

8 [1] Department of Computer Science and Software Engineering, Jinnah University for Women,
9 Karachi, Sindh, Pakistan

11 Corresponding Author:
12 Hafiza Anisa Ahmed[1]
13 5C, Nazimabad, Karachi – 74600, Pakistan
14 Email address: hafizaanisaahmed@gmail.com

Abstract

18 The expeditious growth of the World Wide Web and the rampant flow of network traffic have
19 resulted in a continuous increase of network security threats. Cyber attackers seek to exploit
20 vulnerabilities in network architecture to steal valuable information or disrupt computer resources.
21 Network Intrusion Detection System (NIDS) is used to effectively detect various attacks, thus
22 providing timely protection to network resources from these attacks. To implement NIDS, stream
23 of supervised and unsupervised machine learning approaches is applied to detect irregularities in
24 network traffic and to address network security issues. Such NIDSs are trained using various
25 datasets that include attack traces. However, due to the advancement in modern-day attacks, these
26 systems are unable to detect the emerging threats. Therefore, NIDS needs to be trained and
27 developed with a modern comprehensive dataset which contains contemporary common and attack
28 activities. This paper presents a framework in which different machine learning classification
29 schemes are employed to detect various types of network attack categories. Five machine learning
30 algorithms: Random Forest, Decision Tree, Logistic Regression, K-Nearest Neighbors and
31 Artificial Neural Networks, are used for attack detection. This study uses a dataset published by
32 the University of New South Wales (UNSW-NB15), a relatively new dataset that contains a large
33 amount of network traffic data with nine categories of network attacks. The results show that the
34 classification models achieved the highest accuracy of 89.29% by applying the Random Forest
35 algorithm. Further improvement in the accuracy of classification models is observed when
36 Synthetic Minority Oversampling Technique (SMOTE) is applied to address the class imbalance
37 problem. After applying the SMOTE, the Random Forest classifier showed an accuracy of 95.1%
38 with 24 selected features from the Principal Component Analysis method.

39

Introduction

In today's developed and interconnected world, the number of networks and data security breaches is increasing immensely. The reasons include the growth of network traffic and advances in technology that have led to the creation of newer types of attacks. As a result, the level of attack eventually increases (Mikhail, Fossaceca and Iammartino, 2019).

There exist numerous network security attacks in today's era and to timely detect these attacks, several NIDSs are being developed and deployed. These NIDSs are widely used to protect digital resources against attacks and intrusions on networks (Zong, Chow and Susilo, 2018; Vidal and VidalMonge, 2019). Intrusion detection systems use two different methods, that is, anomaly-based detection and signature-based detection (Moustafa, Creech and Slay, 2017; Li *et al.*, 2019). In an anomaly detection system, the network traffic is monitored and critical network characteristics are continuously tracked and analyzed (Habeeb *et al.*, 2018). It generates alerts if unusual or anomalous activity is detected. Whereas, in signature detection system, well-known patterns of attacks (known as signatures) are stored. The network packets are searched for those patterns (Faker and Dogdu, 2019). If a pattern is accurately matched, the system generates an alert regarding that malicious activity (Azeez *et al.*, 2020).

Although various attack detection mechanisms are available, they are still not effective enough to detect and analyze intrusions or malicious activities (Ahmed, Mahmood and Hu, 2016). Typically, anomaly-based detection systems are developed using different machine learning techniques for predicting intrusions in networks. Research has been conducted in this regard using datasets such as KDDCUP99 (Choudhary and Kesswani, 2020), KDD98 (W. Haider *et al.*, 2017) and NSL-KDD7 (Rathore and Park, 2018). However, due to the evolution of computer networks, these datasets are negatively affecting the results of NIDS (Khraisat *et al.*, 2019). One of the factors influencing the results is the availability of modern-day attack data, as these datasets were created almost two decades ago. Consequently, due to revolution of network traffic, the traffic data available in those datasets is different from the existing modern-day traffic (Moustafa and Slay, 2016).

Improving the performance of existing NIDS requires modern and state-of-the-art datasets that are up-to-date. Therefore, a more efficient and more accurate evaluation of NIDS requires relatively new state-of-the-art datasets, including modern-day network's normal and attack activities. In this research, a framework has been developed for attack detection in a network using the UNSW-NB15 dataset (Tama and Rhee, 2019; Aissa, Guerroumi and Derhab, 2020). This dataset is more recent and includes new attacks. Previously, KDDCUP99, KDD98, and NSL-KDD7 were widely used for NIDS benchmarked datasets. However, these older datasets are not as useful for today's network traffic (Viet *et al.*, 2018; Kumar *et al.*, 2020). However, few researchers have used the new data set, the UNSW-NB15 dataset, to detect an attack but their work has been limited (Bagui *et al.*, 2019).

80   The model proposed in this study uses the UNSW-NB15[1] dataset and not only achieves better
81   accuracy than previous research (Kasongo and Sun, 2020; Kumar, Das and Sinha, 2021) but also
82   effectively detects all categories of attack. The work has been done using Python. Initially, pre-
83   processing techniques were used on 80,000 randomly selected instances from the UNSW-NB15
84   dataset to normalize data values. Later, feature engineering has been performed to select the
85   relevant features. To improve the performance of the classifiers, the research solved the problem
86   of class imbalance using SMOTE. Subsequently, Random Forest (RF), Decision Tree (DT),
87   Logistic Regression (LR), K-Nearest Neighbors (KNN) and Artificial Neural Network (ANN)
88   have been used for classification. Lastly, evaluation metrics were used to compare the performance
89   of all classifiers.
90
91   Following are the major contributions of this research:
92   •   The dataset includes 45 features from which we identified 24 features that were most
93   significant in identifying the attack.
94   •   The various pre-processing techniques have been collectively applied to the UNSW-NB15
95   dataset to make the data meaningful and informative for model training.
96   •   The class imbalance problem is addressed using when Synthetic Minority Oversampling
97   Technique (SMOTE), thereby improving the detection rate of rare attacks.
98   •   We have provided a comparison of five machine learning algorithms for detecting network
99   attack categories.
100
101 The rest of the paper is organized as follows: In Section 2, related work has been presented; Section
102 3 describes the methodology of the framework developed; Section 4 and 5 elaborates the
103 discussion of the experimental results and the last Section 6 hence concludes the paper.
104

Related Work

106 As technology advances with modern techniques, computer networks are using the latest
107 technologies to put it into practice, which has dramatically changed the level of attacks. Therefore,
108 to target the present-day attack categories, UNSW-NB15 dataset has been created (Moustafa and
109 Slay, 2015; Viet *et al.*, 2018)
110
111 The research conducted using the UNSW-NB15 dataset is still not sufficient. However, some of
112 the research work done using datasets is discussed below. Table 1 presents the summary and
113 comparison of the discussed related work.
114
115 Moustafa and Slay (Moustafa and Slay, 2015) developed a model that focused on the classification
116 of attack families available in the UNSW-NB15 dataset. The study used the Association Rule
117 Mining technique for feature selection. For classification, Expectation–Maximization (EM)

---

[1] https://research.unsw.edu.au/projects/unsw-nb15-dataset

118   algorithm and NB have been used. However, the accuracy of both algorithms for detecting rare
119   attacks was not significantly higher as the Naïve Bayes had an accuracy of 78.06% and the
120   accuracy of EM was 58.88%.
121

122   Moustafa and Slay (Moustafa and Slay, 2016) further extended their work in 2016 and used
123   correlation coefficient and gain ratio for feature selection in their work. Thereafter, five
124   classification algorithms of NB, DT, ANN, LR, and EM were used on the UNSW-NB15. Results
125   showed that 85% accuracy was achieved using DT with 15.75 False Alarm Rate (FAR). This
126   research utilized a subset of UNSW-NB15; however, detection accuracy was not satisfactory.
127

128   For detecting botnets and their tracks, Koroniotis et al. (Koroniotis *et al.*, 2017) presented a
129   framework using machine learning techniques on a subset of the UNSW-NB15 dataset using
130   network flow identifiers. Four classification algorithms were used i.e., Association Rule Mining
131   (ARM), ANN, NB and DT. The results showed that the DT obtained the highest accuracy of
132   93.23% with a False Positive Rate (FPR) of 6.77%.
133

134   In 2019, Meftah et al. (Meftah, Rachidi and Assem, 2019) applied a two-stage anomaly-based
135   NIDS approach to detect network attacks. The proposed method used LR, Gradient Boost Machine
136   (GBM) and Support Vector Machine (SVM) with the Recursive Feature Elimination (RFE) and
137   RF feature selection techniques on a complete UNSW-NB15 dataset. The results showed that the
138   accuracy of multi-classifiers using DT was approximately 86.04%, respectively.
139

140   Kumar et al. (Kumar *et al.*, 2020) proposed an integrated calcification-based NIDS using DT
141   models with a combination of clusters created using the k-mean algorithm and IG's feature
142   selection technique. The research utilized only 22 features and five types of network attacks of
143   UNSW-NB15 dataset, and the RTNITP18 dataset, which served as a test dataset to test the
144   performance of the proposed model. The result showed an accuracy of 84.83% using the proposed
145   model and 90.74% using the C5 model of DT.
146

147   Kasongo and Sun (Kasongo and Sun, 2020) presented the NIDS approach using five classification
148   algorithms of LR, KNN, ANN, DT and SVM in conjunction with the feature selection technique
149   of the XGBoost algorithm. The research used the UNSW-NB15 dataset to apply binary and
150   multiclass classification methods. Although binary classification performed well with an accuracy
151   of 96.76% using the KNN classifier, multiclass classification didn't perform well as it achieved the
152   highest accuracy of 82.66%.
153

154   Kumar et al. (Kumar, Das and Sinha, 2021) proposed Unified Intrusion Detection System (UIDS)
155   to detect normal traffic and four types of network attack categories by utilizing UNSW-NB15
156   dataset. Proposed UIDS model was designed with the set of rules (R) derived from various DT
157   models including k-means clustering and IG's feature selection technique. In addition, various

158 algorithms such as C5, Neural Network and SVM were also used to train the model. As a result,
159 the proposed model improved with an accuracy of 88.92% over other approaches. However, other
160 algorithms such as C5, Neural Network and SVM achieved an accuracy of 89.76%, 86.7% and
161 78.77%, respectively.

162

163 From a brief review of related literature as shown in Table 1, it is evident that more work needs to
164 be done to identify the features for the families of network attacks. There is a need to determine a
165 generic model that provides better accuracy for all the attacks presented in the dataset.

166

167 This research provides a model that determines a common subset of features. Subsequently, by
168 using that feature subset we would be able to identify all attacks, belonging to any category with
169 consistent accuracy. It focuses on the implementation of a generic model that provides improved
170 classification accuracy. Moreover, there is limited research that has used the class imbalance
171 technique to balance instances of rare attacks present in the dataset.

172

173 ## Proposed Methodology

174 The framework utilizes a subset of the UNSW-NB15 dataset. It consists of two main steps. The
175 first step involves data pre-processing, in which standardization and normalization of data are
176 performed. Due to the high dimensional nature of the dataset, some features that are irrelevant or
177 redundant may lead to reduce the accuracy of attack detection. To solve this problem, feature
178 selection is used, in which only the relevant subset of features is selected to eliminate useless and
179 noisy features from multidimensional datasets. Afterward, we have then addressed the class
180 imbalance problem. In the next step, different classifiers are trained with relevant features to detect
181 all categories of attack to get maximum accuracy. Finally, accuracy, precision, recall and F1-score
182 performance measures are used to evaluate the model. The proposed methodology that represents
183 the overall framework is shown in Fig. 1.

184

185 **Dataset**
186 UNSW-NB15 dataset has been created by researchers in 2015 focusing on advanced network
187 intrusion techniques. It contains 2.5 million records with 49 features (Dahiya and Srivastava,
188 2018). There are nine different classes of attack families with two label values i.e., normal or attack
189 (Khan *et al.*, 2019; Khammassi and Krichen, 2020) in the UNSW-NB15 dataset (Benmessahel,
190 Xie and Chellal, 2018). These classes are described in Table 2.

191

192 **Dataset Pre-Processing**
193 This phase involves the following steps: data standardization and data normalization.

194

195 • Data Standardization
196 As there were features with different ranges of values in the dataset. Initially, we performed data
197 standardization to convert the data from normal distribution into standard normal distribution.

198 Therefore, after rescaling, a mean value of an attribute is equal to 0 and the resulting distribution
199 is equal to the standard deviation. The formula to calculate a standard score (z-score) is:

200

201
$$z = \frac{(x - \mu)}{\sigma}$$

202

203 Where x is the data sample, μ is the mean and σ is the standard deviation (Xiao *et al.*, 2019).

204

205 • Data Normalization
206 In data normalization, the value of each continuous attribute is scaled between 0 and 1 such that
207 the result of attributes does not dominate each other (Gupta *et al.*, 2016). In this research, the
208 normalizer class of Python has been used. This class enables the normalization of a particular
209 dataset.

210

211 **Feature Selection**
212 Feature selection is a technique that is used to select features that mostly correlate and contribute
213 to the target variable of the dataset (Aljawarneh, Aldwairi and Yassein, 2017). In this research,
214 feature selection is done using Correlation Attribute Evaluation (CA), Information Gain (IG) and
215 Principal Component Analysis (PCA). CA measures the relationship between each feature with
216 the target variable and select only those relative features that have moderately higher positive or
217 negative values, i.e., closer to 1 or -1 (Sugianela and Ahmad, 2020). While IG feature selection
218 technique is used to determine relevant features and minimizing noise caused by unrelated
219 features. These relevant features are calculated from the entropy matrix which measures the
220 uncertainty of the dataset (Kurniabudi *et al.*, 2020). Through Principal Component Analysis, the
221 size of large datasets is reduced by retaining the relevant features that depend on the target class
222 (Kumar, Glisson and Benton, 2020).

223

224 The above-mentioned feature selection techniques help to train the model correctly with only the
225 relevant features that accurately predict the target class.

226

227 **Class Imbalance**
228 The UNSW-NB15 dataset is highly imbalanced not only because the number of normal traffic
229 instances is much higher than different attack categories, but also because the different categories
230 of attack instances are not equal in distribution. This problem is known as "Class Imbalance".

231

232 Table 3 depicts the distribution of nine categories of attack and normal instances in the training
233 dataset. The attack categories such as Analysis, Backdoor, Shellcode and Worms have very few
234 instances. This highly imbalanced nature of the dataset causes problems in training machine
235 learning algorithms for accurately predicting cyber-attacks. To address the class imbalance issue,
236 this research uses SMOTE. SMOTE synthesizes instances of minority classes to balance all the

237   classes in the dataset (Laureano, Sison and Medina, 2019). Table 4 shows the instance percentages
238   in each class after applying SMOTE.
239
240   **Classification Algorithms**
241   Five classification algorithms, that is, RF, DT, LR, KNN and ANN were employed to train the
242   model.
243
244   •   Random Forest
245   Random Forest is an ensemble classifier that is used for improving classification results. It
246   comprises multiple Decision Trees. In comparison with other classifiers, RF provides lower
247   classification errors. Randomization is applied for the selection of the best nodes for splitting when
248   creating separate trees in RF (Jiang *et al.*, 2018).
249
250   •   Decision Tree
251   In the Decision Tree algorithm, the attributes are tested on internal nodes, the outcomes of the tests
252   are represented by branches, and leaf nodes hold labels of the classes (Afraei, Shahriar and Madani,
253   2019). Attribute selection methods are used for identifying nodes. Those selected attributes
254   minimize the information that is required for tuple classification in the resulted partition. Hence,
255   reflecting the minimum uncertainty or impurity in those partitions. Therefore, minimizing the
256   projected number of tests required for tuple classification. In this research, ID3 algorithms utilize
257   entropy class to determine which attributes should be queried on, at every node of those decision
258   trees.
259
260   •   Logistic Regression
261   Logistic Regression is a probabilistic classification model. It casts the problem into a generalized
262   linear regression form. It has a sigmoid curve. The equation of the sigmoid function or logistic
263   function is:

264
$$S(x) = \frac{e^x}{1 + e^x}$$

265
266   This function is used for mapping values to probabilities. It works by mapping real values to other
267   values between 0 and 1 (Kyurkchiev and Markov, 2016).
268
269   •   K-Nearest Neighbors
270   In K-Nearest Neighbors, a new data point is attached with the data points in the training set and
271   based on that attachment, a value is assigned to that new data point. This uses feature similarity
272   for prediction. In KNN Euclidean, Manhattan or Hamming distance are used for calculating the
273   distance between a test data and each record of training data (Jain, Jain and Vishwakarma, 2020).
274   Afterward, according to the value of distance, the rows are sorted. From those rows, K rows from

275 the top are selected. Based on the most frequent classes of these rows classes to the test points are
276 assigned.
277
278 • Artificial Neural Network
279 In the Artificial Neural Network algorithm, there are three layers that consist of computational
280 units called neurons. These layers are input, output and hidden layers. The number of neurons in
281 these layers depends on the features of the dataset and classes which have to be detected and chosen
282 with different techniques. Different types of activation functions are used in the ANN algorithm
283 for calculating the weighted sum of the connections between neurons. This algorithm has biases
284 in the hidden layer and an output layer which are adjusted to reduce errors and improve accuracy
285 in training and testing the model (Andropov *et al.*, 2017).
286
287 **Evaluation Metrics**
288 A confusion matrix is used for the comparison of the performance of machine learning algorithms.
289 This matrix is used for the creation of different metrics by the combination of the values of True
290 Negative (TN), True Positive (TP), False Negative (FN) and False Positive (FP) (Tripathy,
291 Agrawal and Rath, 2016). Below are some of the performance measures to evaluate models by the
292 use of the confusion matrix.
293
294 Accuracy shows the correctness or closeness of the approximated value to the actual or true value
295 of the model which means a portion of the total samples that are classified correctly (Lin, Ye and
296 Xu, 2019). The following formula is used to calculate the accuracy of the model:
297

298
$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}$$

299
300 Precision shows which portion of relevant instances is actually positive among the selected
301 instances (Roy and Cheung, 2018). The following formula is used to calculate precision:
302

303 $Precision = \frac{TP}{TP + FP}$

304
305 Recall or True Positive Rate (TPR) calculates the fraction of actual positives that are correctly
306 identified (Ludwig, 2017). The formula used to find recall is:
307

308
$$Recall = \frac{TP}{TP + FN}$$

309 F1-score is interpreted as the harmonic mean of precision and recall means it combines the
310 weighted average of precision and recall (Niyaz *et al.*, 2016). The following formula is used to
311 calculate F1-score:

312

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

314

Experiment and Result Analysis

Following the methodology depicted in Fig. 1, experimental setup is established. In this research, a sample of 80000 instances is randomly selected from the UNSW-NB15 dataset. Initially, data standardization and normalization have been performed to rescale data values of the dataset and then three feature selection techniques are applied to select the most relevant features. Afterward, the class imbalance problem is resolved using SMOTE. Lastly, five classification algorithms i.e., RF, DT, LR, KNN and ANN are used to classify between the attack categories and normal traffic.

Performance Analysis of Classification Models without Feature Selection

Table 5 summarizes the results of classification models without feature selection, in which RF achieved the highest accuracy, precision, recall, and F1-score values approximately 89.29%, 76.9%, 72.6% and 74.1% respectively. Whereas, LR obtained the lowest accuracy, precision, recall, and F1-score values of approximately 82.07%, 50.5%, 42.6% and 42.3%. DT achieved an accuracy of 88.29% with 70.5% precision, 72.7% recall and 71.4% F1-score. KNN also recorded the same accuracy as LR but with better precision, recall and F1-score results. Whereas, ANN recorded average performance with accuracy, precision, recall, and F1-score values of approximately 84.80%, 61.0%, 53.2% and 54.5%.

Performance Analysis of Classification Models with Feature Selection

Three feature selection techniques i.e., CA, IG and PCA are used in this research.

The following 35 features are selected by applying IG: sttl, ct_state_ttl, ct_flw_http_mthd, sbytes, id, smean, sload, dur, sinpkt, rate, proto, ct_dst_src_ltm, service, dbytes, sjit, ct_srv_dst, dload, dinpkt, dmean, ct_srv_src, synack, tcprtt, ct_dst_sport_ltm, djit, ct_src_dport_ltm, dtcpb, stcpb, spkts , dloss , ct_dst_ltm, ackdat, label, dpkts, ct_src_ltm, sloss.

After applying CA method, 24 features have been achieved from the set of 49 features: id, ct_dst_sport_ltm, ct_dst_src_ltm, ct_src_dport_ltm, sttl, ct_srv_dst, ct_srv_src, ct_dst_ltm, ct_src_ltm, ct_state_ttl, state, swin, dwin, proto, service, rate, dttl, stcpb, dtcpb, dmean, dload, tcprtt, ackdat, synack.

By applying PCA, 151 subsets from the set of 49 features were resulted, out of which 10 subsets with 80% results were selected. After tremendous analysis and evaluation of these 10 subsets, 24 features have been extracted. Afterward, five classifiers were trained by using these features: id, dur, dwin, proto, djit, swin, smean, state, service, ct_src_dport_ltm, dbytes, ct_dst_ltm,

350   ct_dst_sport_ltm, ct_src_ltm, dloss, ct_flw_http_mthd, ct_srv_dst, dpkts, sttl, dmean, spkts,
351   sbytes, sloss and sinpkt.

352

353   After training the classifiers with the above features, the results showing in Table 6 were obtained.

354

355   It is observed using IG technique, that the RF classifier achieved the highest accuracy of 89.5%
356   approx. with precision rate (76.8%), recall (72.3%) and F1-score (73.7%). In contrast to other
357   classifiers, LR and KNN didn't perform well with IG as their recall and F1-score have below 50%
358   scores. There is no much difference between the accuracy of RF and DT classifiers as both give
359   almost the same accuracy, recall and F1-score measures using IG technique. The only difference
360   is in the precision rate as RF achieved 76.8% and DT scored 69.6% precision value.

361

362   It is observed that the accuracy of all the classifiers decreased when the model is trained using the
363   CA technique. RF classifier achieved the highest accuracy of 86.3% but with low precision, recall
364   and F1-score measures. The accuracy of DT and ANN classifiers are approximately the same as
365   the RF classifier with a minor difference of 2% to 5%. However, ANN classifier has very low-
366   performance measures as compared to RF and DT. Also, LR and KNN have the lowest accuracy
367   measures with poor performance metrics.

368

369   It is observed using PCA feature selection technique, that RF classifier obtained the highest
370   accuracy of 89.3% with precision (77.3%), recall (70.8%) and F1-score (73.1%) rates. All the
371   classifiers achieved the accuracy in between 80% to 89% but with low performance measures as
372   compared to IG feature selection technique. LR recorded the lowest recall rate with 40.6% and F1-
373   score with 40%.

374

375   After evaluation of the performance of three feature selection methods, it was observed that the
376   feature selection technique of IG and PCA performed well as compared to CA. RF and DT
377   classifiers approximately achieved the same accuracy between 88% to 89% when trained with IG
378   and PCA. However, for precision, recall and F1-score measures, these classifiers showed average
379   scores. Therefore, it is concluded that, no major changes have been observed in the results after
380   applying feature selection techniques as classifiers achieved almost same accuracy before feature
381   selection.

382

383   **Performance Analysis of Classification Models by handling Imbalanced Data**
384   To handle imbalanced data, SMOTE technique has been applied in this research to adjust the class
385   distribution of dataset and increase the instances of minority classes of those network attacks that
386   has lower instances. After handling imbalance data, the results showing in Table 7 were obtained.

387

388   By using IG feature selection technique after applying SMOTE, it is observed that the RF classifier
389   achieved highest accuracy of 95.1% with highest precision rate (94.8%), recall (95.7%) and F1-

390   score (95.1%). Also, the accuracy of DT is 94.7%, almost nearest from the RF. The accuracy of
391   both algorithms increased after handling imbalanced classes i.e., from 89.5% to 95.0% in RF and
392   88.5% to 94.5% in DT. Whereas, after applying SMOTE, LR and ANN didn't perform well as
393   their accuracies were decreased from 82.2% to 69.4% in LR and 85.7% to 77.3% in ANN using
394   IG method. The accuracy of KNN is almost the same using all three feature selection techniques
395   but with good precision, recall and F1-score measures.
396
397   By using CA feature selection technique after applying SMOTE, it is noticed that RF and DT
398   classifiers achieved highest accuracy in between 92.6% to 93.5% with above 90% precision, recall
399   and F1-score measures. The accuracy of both the algorithms increased after applying SMOTE.
400   Also, a minor change occurred in KNN as their accuracy is improved from 76.8% to 78.4% after
401   handing imbalanced classes. However, when comes to LR and ANN, both algorithms did not
402   perform well with class balance as their accuracies have been decreased i.e., in LR, from 74.5% to
403   62.0% and in ANN, from 80.3% to 71.7%.
404
405   By using PCA after applying SMOTE, it is noticed that the RF classifier achieved the highest
406   accuracy of 95.1% with a precision rate of 94.8%, recall of 95.7% and F1-score of 95.1%. DT
407   classifier achieved the accuracy of 94.7%, which is almost nearest to the accuracy of RF. After
408   applying SMOTE, there is no change in the results of KNN using PCA method. However, the
409   accuracy of LR and ANN decreased from 80.4% to 68.2% in LR and 85.2% to 77.6% in ANN but
410   with increased precision, recall and F1-score measures.
411
412   **Overall Performance Evaluation of Classification Models after handling Class Imbalance**
413
414   After handling class balancing by using SMOTE, it is concluded that RF classifier performed well
415   with good results up to 95.1% by using PCA feature selection technique. Also, it is noticed that
416   class balancing did not impact on LR and ANN classifiers as their accuracy decreased after
417   handling minority classes.
418
419   • Confusion Metrics of best performed Classifier: Random Forest
420   After analysis of the five classification models, it is observed that RF scheme provided the highest
421   accuracy. On the basis of which, the confusion matrix of RF classification model is analyzed to
422   observe the attack prediction accuracy of the nine categories of attacks separately.
423
424   In Fig. 2, it is depicted that all the normal traffic instances were identified correctly by RF (i.e., it
425   had 100% accuracy). In attack categories, all the instances of Backdoor, Shellcode and Worms
426   were also identified correctly showing 100 prediction accuracy. Whereas, 1759 out of 1763
427   instances of Analysis attack (i.e., 99.77% accuracy), 2341 out of 2534 instances of Fuzzers (i.e.,
428   92.38% accuracy), 5461 out of 5545 instances of Generic (i.e., 98.49% accuracy), 2151 out of
429   2357 instances of Reconnaissance (i.e., 91.26% accuracy) were identified correctly.

430

431

Discussion

432

433   The research proposed a framework that predicts a variety of network attack categories using
434   supervised machine learning algorithms. The dataset used in this study is the UNSW-NB15
435   dataset, a relatively new and containing a large amount of network traffic data, with nine types of
436   network attack categories.

437

438   The proposed framework implies five machine learning algorithms in conjunction with pre-
439   processing techniques, different methods of feature selection and SMOTE. After training, the
440   results of the classifiers shown in Table 6, 7 and 8 were obtained.

441

442   Compared to previous studies, as shown in Table 1, our model performed well with the highest
443   accuracy of 95.1% using RF classifier with 24 features selected by PCA after applying SMOTE.
444   DT classifier has also performed well with accuracy between 92.6% to 94.7% using different
445   feature selection techniques. Existing studies that summarized in Table 1, achieved less than 90%
446   accuracy, except for the research proposed by Kumar et al. (Kumar *et al.*, 2020) and Koroniotis et
447   al. (Koroniotis *et al.*, 2017). Kumar et al. showed 90.74% accuracy using the C5 model of DT in
448   conjunction with the IG feature selection technique. While the model proposed by Koroniotis et
449   al. achieved 93.23% accuracy using DT classifier. In addition, none of the studies listed in Table
450   1 have resolved the class imbalance problem of the UNSW-NB15 dataset as there are many studies
451   (Al-Daweri *et al.*, 2020; Ahmad *et al.*, 2021; Bagui and Li, 2021; Dlamini and Fahim, 2021) that
452   have highlighted this issue. We addressed the class imbalance problem by applying SMOTE that
453   improved the performance of the classifiers and achieved good results.

454

Conclusions

455

456   This paper presents a framework for network intrusion detection. The performance of the proposed
457   framework has been analyzed and evaluated on the UNSW-NB15 dataset. The proposed
458   framework uses different pre-processing techniques that includes data standardization and
459   normalization, feature selection techniques and class balancing methods. The usability of the
460   selected features along with using data standardization and normalization techniques is analyzed
461   by applying them on five different classification models. The results showed that the features
462   selected by PCA contributed much to improve accuracy than other methods. For improving the
463   accuracy of the classification models, the class imbalance problem is also addressed which
464   increased the framework performance with high margin. In can be concluded on the basis of
465   evaluation results that both RF and DT classifiers performed well over the UNSW-NB15 dataset
466   in terms of accuracy, precision, recall, and F1-score metrics. It can also be concluded that major
467   issue in UNSW-NB15 dataset is not only the presence of highly correlated features but also the
468   class imbalance problem of the dataset. Therefore, we used a novel combination of different pre-

469   processing techniques in order to resolve all the underlying issues of the dataset and developed a
470   fast and efficient network security intrusion detection system.

471

References

473   Afraei, S., Shahriar, K. and Madani, S. H. (2019) 'Developing intelligent classification models for
474   rock burst prediction after recognizing significant predictor variables, Section 2: Designing
475   classifiers', in *Tunnelling and Underground Space Technology*. Elsevier, pp. 522–537. doi:
476   https://doi.org/10.1016/j.tust.2018.09.022.

477   Ahmad, M. *et al.* (2021) 'Intrusion detection in internet of things using supervised machine
478   learning based on application and transport layer features using UNSW-NB15 data-set', *Eurasip
479   Journal on Wireless Communications and Networking*, (1). doi: 10.1186/s13638-021-01893-8.

480   Ahmed, M., Mahmood, A. N. and Hu, J. (2016) 'A survey of network anomaly detection
481   techniques', *Journal of Network and Computer Applications*, 60, pp. 19–31.

482   Aissa, N. B., Guerroumi, M. and Derhab, A. (2020) 'NSNAD : negative selection-based network
483   anomaly detection approach with relevant feature subset', *Neural Computing and Applications*,
484   32(8), pp. 3475–3501. doi: https://doi.org/10.1007/s00521-019-04396-2.

485   Al-Daweri, M. S. *et al.* (2020) 'An analysis of the KDD99 and UNSW-NB15 datasets for the
486   intrusion detection system', *Symmetry*, 12(10), pp. 1–32. doi: 10.3390/sym12101666.

487   Aljawarneh, S., Aldwairi, M. and Yassein, M. B. (2017) 'Anomaly-based intrusion detection
488   system through feature selection analysis and building hybrid efficient model', *Journal of
489   Computational Science*, 25, pp. 152–160. doi: https://doi.org/10.1016/j.jocs.2017.03.006.

490   Andropov, S. *et al.* (2017) 'Network Anomaly Detection using Artificial Neural Networks', in
491   *2017 20th Conference of Open Innovations Association (FRUCT)*. Saint Petersburg, Russia: IEEE,
492   pp. 26–31. doi: 10.23919/FRUCT.2017.8071288.

493   Azeez, N. A. *et al.* (2020) 'Intrusion Detection and Prevention Systems: An Updated Review', in
494   *Data Management, Analytics and Innovation*. Springer Singapore, pp. 685–696. doi:
495   https://doi.org/10.1007/978-981-32-9949-8_48.

496   Bagui, S. and Li, K. (2021) 'Resampling imbalanced data for network intrusion detection datasets',
497   *Journal of Big Data*, 8(1). doi: 10.1186/s40537-020-00390-x.

498   Bagui, Sikha *et al.* (2019) 'Using machine learning techniques to identify rare cyber-attacks on the
499   UNSW-NB15 dataset', *Security and Privacy*, 2(6), pp. 1–13. doi: https://doi.org/10.1002/spy2.91.

500   Benmessahel, I., Xie, K. and Chellal, M. (2018) 'A new evolutionary neural networks based on
501   intrusion detection systems using multiverse optimization', *Applied Intelligence*, 48(8), pp. 2315–
502   2327. doi: https://doi.org/10.1007/s10489-017-1085-y.

503   Choudhary, S. and Kesswani, N. (2020) 'Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15
504   Datasets using Deep Learning in IoT', *Procedia Computer Science*, 167(2019), pp. 1561–1573.
505   doi: https://doi.org/10.1016/j.procs.2020.03.367.

506   Dahiya, P. and Srivastava, D. K. (2018) 'Network Intrusion Detection in Big Dataset Using Spark',
507   *Procedia Computer Science*, 132, pp. 253–262. doi: https://doi.org/10.1016/j.procs.2018.05.169.

508   Dlamini, G. and Fahim, M. (2021) 'DGM: a data generative model to improve minority class
509   presence in anomaly detection domain', *Neural Computing and Applications*, 33(20), pp. 13635–
510   13646. doi: 10.1007/s00521-021-05993-w.

511   Faker, O. and Dogdu, E. (2019) 'Intrusion Detection Using Big Data and Deep Learning
512   Techniques', in *ACM SE '19: Proceedings of the 2019 ACM Southeast Conference*. Kennesaw
513   GA, USA: ACM, pp. 86–93. doi: https://doi.org/10.1145/3299815.3314439.

514   Gupta, D. *et al.* (2016) 'Network intrusion detection system using various data mining techniques',
515   in *2016 International Conference on Research Advances in Integrated Navigation Systems*
516   *(RAINS)*. Bangalore, India: IEEE, pp. 1–6. doi: https://doi.org/10.1109/RAINS.2016.7764418.
517   Habeeb, R. A. A. *et al.* (2018) 'Real-time big data processing for anomaly detection : A Survey',
518   *International Journal of Information Management*, 45, pp. 289–307. doi:
519   https://doi.org/10.1016/j.ijinfomgt.2018.08.006.
520   Jain, S., Jain, S. C. and Vishwakarma, S. (2020) 'Analysis and Prediction of Customers' Reviews
521   with Amazon Dataset on Products', in *ICT Systems and Sustainability*. Singapore: Springer, pp.
522   445-456. doi: https://doi.org/10.1007/978-981-15-0936-0_48.
523   Jiang, J. *et al.* (2018) 'RST-RF : A Hybrid Model based on Rough Set Theory and Random Forest
524   for Network Intrusion Detection', in *ICCSP 2018: Proceedings of the 2nd International*
525   *Conference on Cryptography, Security and Privacy*. Guiyang, China: ACM, pp. 77–81. doi:
526   https://doi.org/10.1145/3199478.3199489.
527   Kasongo, S. M. and Sun, Y. (2020) 'Performance Analysis of Intrusion Detection Systems Using
528   a Feature Selection Method on the UNSW-NB15 Dataset', *Journal of Big Data*, 7(1). doi:
529   10.1186/s40537-020-00379-6.
530   Khammassi, C. and Krichen, S. (2020) 'A NSGA2-LR Wrapper Approach for Feature Selection
531   in Network Intrusion Detection', *Computer Networks*, 172(March), p. 107183. doi:
532   https://doi.org/10.1016/j.comnet.2020.107183.
533   Khan, F. A. *et al.* (2019) 'A Novel Two-Stage Deep Learning Model for Efficient Network
534   Intrusion Detection', *IEEE Access*, 7, pp. 30373–30385. doi:
535   https://doi.org/10.1109/ACCESS.2019.2899721.
536   Khraisat, A. *et al.* (2019) 'Survey of intrusion detection systems : techniques , datasets and
537   challenges', *Cybersecurity*, 2(1), pp. 2–20. doi: https://doi.org/10.1186/s42400-019-0038-7.
538   Koroniotis, N. *et al.* (2017) 'Towards Developing Network forensic mechanism for Botnet
539   Activities in the IoT based on Machine Learning Techniques', in *International Conference on*
540   *Mobile Networks and Management*. Springer, Cham., pp. 30–44.
541   Kumar, A., Glisson, W. B. and Benton, R. (2020) 'Network attack detection using an unsupervised
542   machine learning algorithm', in *Proceedings of the Annual Hawaii International Conference on*
543   *System Sciences*, pp. 6496–6505. doi: 10.24251/hicss.2020.795.
544   Kumar, V. *et al.* (2020) 'An integrated rule based intrusion detection system : analysis on UNSW-
545   NB15 data set and the real time online dataset', *Cluster Computing*, 23, pp. 1–22. doi:
546   https://doi.org/10.1007/s10586-019-03008-x.
547   Kumar, V., Das, A. K. and Sinha, D. (2021) 'UIDS: a unified intrusion detection system for IoT
548   environment', *Evolutionary Intelligence*, 14(1), pp. 47–59. doi: 10.1007/s12065-019-00291-w.
549   Kurniabudi *et al.* (2020) 'CICIDS-2017 Dataset Feature Analysis with Information Gain for
550   Anomaly Detection', *IEEE Access*, 8, pp. 132911–132921. doi: 10.1109/ACCESS.2020.3009843.
551   Kyurkchiev, N. and Markov, S. (2016) 'On the Hausdorff distance between the Heaviside step
552   function and Verhulst logistic function', *Journal of Mathematical Chemistry*, 54(1), pp. 109–119.
553   doi: https://doi.org/10.1007/s10910-015-0552-0.
554   Laureano, L. B., Sison, A. M. and Medina, R. P. (2019) 'Handling Imbalanced Data through
555   Affinity Propagation and SMOTE', in *ICCBD 2019: Proceedings of the 2nd International*
556   *Conference on Computing and Big Data*. Taichung, Taiwan: ACM, pp. 22–26. doi:
557   https://doi.org/10.1145/3366650.3366665.
558   Li, W. *et al.* (2019) 'Designing Collaborative Blockchained Signature-based Intrusion Detection
559   in', *Future Generation Computer Systems*, 96, pp. 481–489. doi:

560 https://doi.org/10.1016/j.future.2019.02.064.

561 Lin, P., Ye, K. and Xu, C. (2019) 'Dynamic Network Anomaly Detection System by Using Deep
562 Learning Techniques', in *CLOUD 2019: International Conference on Cloud Computing*. San
563 Diego, CA, USA: Springer, Cham, pp. 161–176. doi: 10.1007/978-3-030-23502-4.

564 Ludwig, S. A. (2017) 'Intrusion Detection of Multiple Attack Classes using a Deep Neural Net
565 Ensemble', in *IEEE Symposium Series on Computational Intelligence (SSCI)*. Honolulu, HI, USA:
566 IEEE, pp. 1–7. doi: https://doi.org/10.1109/SSCI.2017.8280825.

567 Meftah, S., Rachidi, T. and Assem, N. (2019) 'Network based intrusion detection using the
568 UNSW-NB15 dataset', *International Journal of Computing and Digital Systems*, 8(5), pp. 477–
569 487. doi: 10.12785/ijcds/080505.

570 Mikhail, J. W., Fossaceca, J. M. and Iammartino, R. (2019) 'A Semi-Boosted Nested Model With
571 Sensitivity-Based Weighted Binarization for Multi-Domain Network Intrusion Detection', *ACM*
572 *Transactions on Intelligent Systems and Technology*, 10(3), pp. 1–27. doi:
573 https://doi.org/10.1145/3313778.

574 Moustafa, N., Creech, G. and Slay, J. (2017) 'Anomaly Detection System using Beta Mixture
575 Models and Outlier Detection', in *Progress in Computing, Analytics and Networking. Advances in*
576 *Intelligent Systems and Computing*. Springer, Singapore, pp. 125–135. doi:
577 https://doi.org/10.1007/978-981-10-7871-2_13.

578 Moustafa, N. and Slay, J. (2015) 'The significant features of the UNSW-NB15 and the KDD99
579 data sets for Network Intrusion Detection Systems', in *4th international workshop on building*
580 *analysis datasets and gathering experience returns for security (BADGERS)*. IEEE, pp. 25–31.

581 Moustafa, N. and Slay, J. (2016) 'The evaluation of Network Anomaly Detection Systems :
582 Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set',
583 *Information Security Journal A Global Perspective*, 25(1–3), pp. 18–31.

584 Niyaz, Q. *et al.* (2016) 'A Deep Learning Approach for Network Intrusion Detection System', in
585 *Proceedings of the 9th EAI International Conference on Bio-inspired Information and*
586 *Communications Technologies (BIONETICS)*. New York, USA: ACM, pp. 21–26. doi:
587 https://doi.org/10.4108/eai.3-12-2015.2262516.

588 Rathore, S. and Park, J. H. (2018) 'Semi-supervised learning based distributed attack detection
589 framework for IoT', *Applied Soft Computing Journal*, 72, pp. 79–89. doi:
590 https://doi.org/10.1016/j.asoc.2018.05.049.

591 Roy, B. and Cheung, H. (2018) 'A Deep Learning Approach for Intrusion Detection in Internet of
592 Things using Bi-Directional Long Short-Term Memory Recurrent Neural Network.', in *28th*
593 *International Telecommunication Networks and Applications Conference (ITNAC)*. Sydney,
594 NSW, Australia: IEEE, pp. 1–6. doi: https://doi.org/10.1109/ATNAC.2018.8615294.

595 Sugianela, Y. and Ahmad, T. (2020) 'Pearson Correlation Attribute Evaluation-based Feature
596 Selection for Intrusion Detection System', in *International Conference on Smart Technology and*
597 *Applications (ICoSTA)*. Surabaya, Indonesia: IEEE. doi:
598 10.1109/ICoSTA48221.2020.1570613717.

599 Tama, B. A. and Rhee, K.-H. (2019) 'An in-depth experimental study of anomaly detection using
600 gradient boosted machine', *Neural Computing and Applications*, 31(4), pp. 955–965. doi:
601 https://doi.org/10.1007/s00521-017-3128-z.

602 Tripathy, A., Agrawal, A. and Rath, S. K. (2016) 'Classification of Sentiment Reviews using N-
603 gram Machine Learning Approach', *Expert Systems With Applications*, 57, pp. 117–126. doi:
604 https://doi.org/10.1016/j.eswa.2016.03.028Get rights and content.

605 Vidal, J. M. and VidalMonge, M. A. S. (2019) 'Adversarial Communication Networks Modeling

606   for Intrusion Detection Strengthened against Mimicry', in *ARES '19:Proceedings of the 14th*
607   *International Conference on Availability, Reliability and Security*. Canterbury CA, United
608   Kingdom: ACM, pp. 1–6. doi: https://doi.org/10.1145/3339252.3340335.

609   Viet, H. N. *et al.* (2018) 'Using Deep Learning Model for Network Scanning Detection', in *ICFET*
610   *'18: Proceedings of the 4th International Conference on Frontiers of Educational Technologies*.
611   Moscow, Russia: ACM, pp. 117–121. doi: https://doi.org/10.1145/3233347.3233379.

612   W. Haider *et al.* (2017) 'Generating realistic intrusion detection system dataset based on fuzzy
613   qualitative modeling', *Journal of Network and Computer Applications*, 87, pp. 185–192. doi:
614   https://doi.org/10.1016/j.jnca.2017.03.018.

615   Xiao, Y. *et al.* (2019) 'An Intrusion Detection Model Based on Feature Reduction and
616   Convolutional Neural Networks', *IEEE Access*, 7, pp. 42210–42219. doi:
617   10.1109/ACCESS.2019.2904620.

618   Zong, W., Chow, Y.-W. and Susilo, W. (2018) 'A two-stage classifier approach for network
619   intrusion detection', in *ISPEC 2018: International Conference on Information Security Practice*
620   *and Experience*. Tokyo, Japan: Springer, Cham, pp. 329–340. doi: https://doi.org/10.1007/978-3-
621   319-99807-7_20.

622

Figure 1

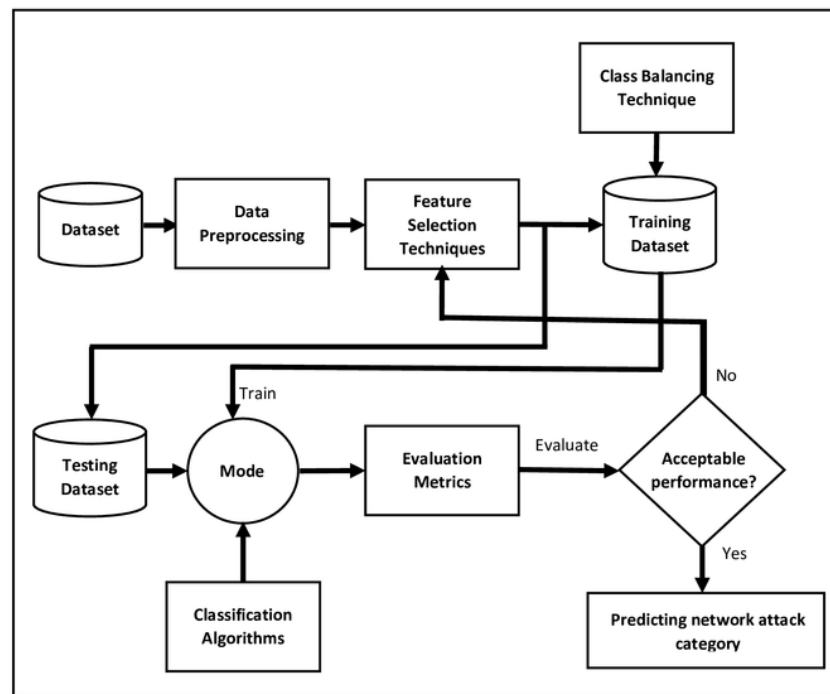
Overall framework for predicting network attack categories

Overall framework for predicting network attack categories

Figure 2

Confusion Matrix of Random Forest Classifier

Predicted		0	1	2	3	4	5	6	7	8	9
True											
Analysis	0	1759	0	2	2	0	0	0	0	0	0
Backdoor	1	0	1762	0	0	0	0	0	0	0	0
DoS	2	39	33	3038	322	60	6	0	36	2	0
Exploits	3	51	33	424	5626	78	8	0	130	14	3
Fuzzers	4	10	13	56	78	2341	0	0	11	21	0
Generic	5	1	3	22	44	9	5461	0	0	0	0
Normal	6	0	0	0	0	0	0	7453	0	0	0
Reconnaissance	7	6	18	64	102	9	2	0	2151	0	0
Shell code	8	0	0	0	0	0	0	0	0	1651	0
Worm	9	0	0	0	0	0	0	0	0	0	1846

Confusion Matrix of Random Forest Classifier

Summary of Existing Studies related to Network Attack Categories

Table 1:

Summary of Existing Studies related to Network Attack Categories

Reference	Dataset (complete /Partial)	Algorithms	Accuracy/ FAR/FPR	Limitations
Moustafa and Slay, 2015	UNSW-NB15 (Partial) KDD99 Dataset	Naïve Bayes and EM Algorithm	Accuracy: Naïve Bayes – 37.5% EM Algorithm: 75.80% FPR: 22.08	This research is determining only five network attack categories. The problem of class imbalance has not been resolved. As a result, the algorithms are not performing well.
Moustafa and Slay, 2016	UNSW-NB15 (Partial) Dataset	Naïve Bayes, Decision Tree, Artificial Neural Network, Logistic Regression, and Expectation-Maximisation	Accuracy: between 78.47% to 85.56% FAR: between 15.75% to 23.79%	In this study, data pre-processing techniques have not been implemented and the issue of class imbalance has not been resolved.
Koroniotis et al., 2017	UNSW-NB15 Dataset	Naive Bayes, Decision Tree, Association Rule Mining (ARM) and Artificial Neural Network	Accuracy: between 63.97% to 93.23% FPR: between 6.77% to 36.03%	This work didn't solve the problem of class imbalance. Hence, the algorithms did not perform well to detect some network attacks.
Meftah, Rachidi and Assem, 2019	UNSW-NB15 Dataset	Logistic Regression, Gradient Boost Machine, and Support Vector Machine	Accuracy: achieving a multi-classification accuracy of 86.04%	This research didn't address the class imbalance problem.
Kumar et al., 2020	UNSW-NB15 (Partial) and RTNITP18 Dataset	Decision Tree Models (C5, CHAID, CART, QUEST)	Accuracy: 84.83% using proposed model and 90.74% using C5 model	This work has predicted only 4 out of 9 categories of the UNSW-NB15 dataset. Also, the problem of class imbalance has not been solved in this study.
Kasongo and Sun,	UNSW-NB15 Dataset	Logistic regression, K-Nearest	Accuracy: between	The problem of class imbalance has not been

		Neighbors, Artificial Neural Network, Decision Tree and Support Vector Machine	53.43% to 82.66% using multiclass classification scheme	resolved in this research. Hence, the model has not achieved good accuracy.
2020				
Kumar et al., 2021	UNSW-NB15 (Partial) Dataset	Decision Tree Models (C5, CHAID, CART, QUEST)	Accuracy: 88.92% using proposed model	Research has predicted only 4 types of network attack categories of the UNSW-NB15 dataset. In addition, the problem of class imbalance has not been resolved in this research.

Description of Network Attacks

Table 1:

Description of Network Attacks

Attack Family	Description
Fuzzers	These attacks attempt to crash servers on networks by inputting numerous random data, called "Fuzz", in vulnerable points of the networks
Analysis	These attacks perform scanning of networks via ports (for example: port scans, footprinting, vulnerability scans).
Backdoors	In this type of attacks intruder bypass normal authentication process of system portals to get illegal access into system. These attacks use malicious software's that provide remote access to the system to the attackers.
Denial of service (DoS)	In this type of attacks, the attackers send several illegal connection requests to generate unwanted network traffic and make network services unavailable for actual users.
Exploits:	In these attacks vulnerable points in the operating systems are targeted and compromised.
Generic	This attack is a collision attack in which attackers tamper secret keys generated by using cryptographic principles
Reconnaissance	These types of attacks try to find possible vulnerabilities in the computer network and then further use different exploitation techniques to exploit a compromised network.
Shellcode	These attacks comprise of a set of instructions that are used as a payload in the exploitation of a certain network. These codes are inserted into software to compromise and remotely access a computer system.
Worms	These are attacks are self-replicating malicious programs that exploit computer systems by duplicating themselves to the uninfected computer systems of the entire network system

Class Distribution in Dataset

1 **Table 1:**

2 **Class Distribution in Dataset**

Class Distribution	Instance count (%)
Normal	32.10
Generic	22.80
Exploits	19.06
Fuzzers	10.32
DoS	7.08
Reconnaissance	5.85
Analysis	1.12
Backdoor	0.93
Shellcode	0.65
Worms	0.06

Class Distribution in Dataset after applying SMOTE

1   **Table 1:**

2   **Class Distribution in Dataset after applying SMOTE**

Class Distribution	Instance count (%)
Normal	22.13
Generic	18.31
Exploits	15.72
Fuzzers	10.05
DoS	7.11
Reconnaissance	6.62
Analysis	5.21
Backdoor	5.08
Shellcode	4.95
Worms	4.76

Results of classification models without feature selection

Table 1:

Results of classification models without feature selection

Method	Accuracy%	Precision%	Recall%	F1-Score%
Random Forest	89.29	76.9	72.6	74.1
Decision Tree	88.29	70.5	72.7	71.4
Logistic Regression	82.07	50.5	42.6	42.3
K-Nearest Neighbor	82.37	52.5	47.6	49.1
Artificial Neural Network	84.80	61.0	53.2	54.5

Results of classification models with feature selection

1 **Table 1:**

2 **Results of classification models with feature selection**

Method	Feature Selection Techniques	Accuracy %	Precision %	Recall %	F1-Score%
Random Forest	Information Gain Attribute	89.5	76.8	72.3	73.7
	Correlation Attribute	86.3	72.2	59.6	61.9
	Principal Component Analysis	89.3	77.3	70.8	73.1
Decision Tree	Information Gain Attribute	88.5	69.6	72.0	70.7
	Correlation Attribute	84.8	63.3	60.9	61.9
	Principal Component Analysis	88.4	70.9	67.3	69.3
Logistic Regression	Information Gain Attribute	82.2	51.2	42.3	41.9
	Correlation Attribute	74.5	38.5	36.0	35.4
	Principal Component Analysis	80.4	51.3	40.6	40.0
K-Nearest Neighbor	Information Gain Attribute	82.7	54.0	48.1	49.6
	Correlation Attribute	76.8	46.3	42.6	43.5
	Principal Component Analysis	84.0	57.8	51.3	53.3
Artificial Neural Network	Information Gain Attribute	85.7	60.6	54.6	54.2
	Correlation Attribute	80.3	50.1	44.1	44.7
	Principal Component Analysis	85.2	61.2	52.4	54.4

Results of classification models with feature selection after handling Class Imbalance

1   **Table 1:**

2   **Results of classification models with feature selection after handling Class Imbalance**

Method	Feature Selection Techniques	Accuracy %	Precision %	Recall %	F1-Score%
Random Forest	Information Gain Attribute	95.0	94.7	95.7	95.1
	Correlation Attribute	93.5	93.0	94.2	93.5
	Principal Component Analysis	95.1	94.8	95.7	95.1
Decision Tree	Information Gain Attribute	94.5	94.1	95.2	94.7
	Correlation Attribute	92.6	91.8	93.4	92.6
	Principal Component Analysis	94.7	94.4	95.4	94.8
Logistic Regression	Information Gain Attribute	69.4	61.0	59.4	56.2
	Correlation Attribute	62.0	48.1	50.6	45.8
	Principal Component Analysis	68.2	57.8	58.0	54.8
K-Nearest Neighbor	Information Gain Attribute	82.7	79.4	82.6	80.6
	Correlation Attribute	78.4	74.3	78.3	75.8
	Principal Component Analysis	84.7	82.2	85.1	83.1
Artificial Neural Network	Information Gain Attribute	77.3	75.0	70.5	71.0
	Correlation Attribute	71.7	69.2	62.1	63.5
	Principal Component Analysis	77.6	76.2	70.6	71.5