Information & Security

M.A. Umar, Chen & Y. Liu
vol. 49 (2021): Preview
https://doi.org/10.11610/isij.4901

Published since 1998 ISSN 0861-5160 (print), ISSN 1314-2119 (online)

Research Article

A Hybrid Intrusion Detection with Decision Tree for Feature Selection

Mubarak Albarka Umar* (✉), *Zhanfang Chen*, *Yan Liu

School of Computer Science and Technology, Changchun University of Science and Technology, 7186 Weixing Road, Jilin, China.

ABSTRACT:

Intrusion detection systems (IDS) typically take high computational complexity to examine data features and identify intrusion patterns due to the size and nature of the current intrusion detection datasets. Data pre-processing techniques (such as feature selection) are being used to reduce such complexity by eliminating irrelevant and redundant features in such datasets. The objective of this study is to analyse the effectiveness and efficiency of some feature selection approaches, namely wrapper-based and filter-based modelling approaches. To achieve that, machine learning models are designed in a hybrid approach with either wrapper or filter selection processes. Five machine learning algorithms are used on the wrapper and filter-based feature selection methods to build the IDS models using the UNSW-NB15 dataset. The wrapper-based hybrid intrusion detection model comprises a decision tree algorithm to guide the selection process and three filter-based methods, namely information gain, gain ratio, and relief, are used for comparison to determine the efficiency and effectiveness of the wrapper approach. Furthermore, a comparison with other state-of-the-art intrusion detection approaches is performed. The experimental results show that the wrapper-based method is quite effective in comparison to state-of-the-art works; however, it requires high computational time in comparison to the filter-based methods while achieving similar results. Our work also revealed unobserved issues on the conformity of the UNSW-NB15 dataset.

ARTICLE INFO:

RECEIVED: 15 APR 2021
REVISED: 29 MAY 2021
ONLINE: 15 JUNE 2021

KEYWORDS:

intrusion detection, hybrid IDS, feature selection, machine learning algorithms, big data, IDS dataset

✉ Tel.: +2349023155878; E-mail: 2018300037@mails.cust.edu.cn

1. Introduction

Today more sophisticated infiltration techniques are being developed by attackers to challenge and defeat the security layer of the internet and computer users. Protecting the confidentiality, credibility, integrity, and availability of information communicated over the internet and across computers has become a vital and challenging task for network security administrators.[1] Thus, an efficient and reliable IDS is needed as an added security layer to the existing less-effective first line of defence solutions to safeguard computer networks from known and unknown vulnerabilities.[2,3] Machine Learning (ML) techniques, due to their ability to learn and improve with experience,[4] are nowadays utilised for building such IDS.[5] However, there was one problem with the initial idea of applying ML in the form of a single classifier in IDS, that is, this approach is not robust enough to build an effective IDS.[6] Thus, to enable building more reliable IDS, researchers have proposed the hybrid IDS modelling approach to enhance the accuracy of detecting an intrusion.[7]

An important aspect of building and validating IDS is the IDS dataset.[8] The dataset typically comes from heterogeneous platforms and can be redundant, incomplete, and inconsistent,[9] which generally affects the detection accuracy and efficiency by increasing computational complexity and expanding the search space of the problem.[10] The primary purpose of IDS is to accurately detect attacks with minimum false alerts. However, to fulfil this purpose, an IDS should be able to handle a huge amount of network data and should be fast enough to allow real-time decisions. Pre-processing techniques such as normalization,[11,12] data filtration,[13] and discretization,[14] among others, are used to overcome such issues. Feature selection is one of these techniques proposed by various researchers [15] and it has notably proven to be the most effective solution.[16]

Feature selection aims to select relevant features and eliminate useless ones with a minimum or no degradation of performance. The feature selection approaches are of three main types, namely filter, wrapper, and embedded approaches.[17] The filter approach extracts features using the general characteristics of the data such as distance, consistency, dependency, information, and correlation without using any learning algorithm in evaluating or selecting feature subsets; it may however result in eliminating relevant and important features. The wrapper approach uses a learning algorithm to determine the most useful and relevant features; in comparison to the filter approach, though computationally more expensive, the wrapper approach improves performance. The embedded approach was proposed to overcome the limitations in filter and wrapper approaches. The embedded approach achieves model fitting and feature selection simultaneously, performing the feature selection during the learning time.[18]

In this article, which is an extension of our previous work,[19] three more filter methods in addition to the wrapper method are compared. The wrapper-based feature selection with decision tree algorithm is first used as a regular means of obtaining an optimal subset of the original features. Then five among the most used ML algorithms in IDS [20] are selected to build the models. The selected algorithms are Artificial Neural Network (ANN), k-Nearest Neighbor (KNN), Support Vector Machine (SVM), Random Forest (RF), and Naïve Bayes (NB). The dataset used for the implementation of the models is the contemporary UNSW-NB15 dataset [21,22] introduced by Moustafa and Slay.[23] One-hot encoding and min-max methods are used for encoding and normalization respectively. In addition to the computation time, the models are evaluated using the three well known IDS evaluation metrics,[24] namely Accuracy, Detection rate, and False alert rate. Furthermore, two comparisons are performed to determine the effectiveness of the methods. Firstly, the four applied feature selection methods (namely, decision tree wrapper-based, information gain (InfoGain), gain ratio (GainRatio), and Relief filters) are compared. Then we compare the best performing model against state-of-the-art works.

The rest of the article is organised as follows: Section 2 introduces the feature selection approach; justification of the proposed feature selection is also given along with an explanation of the three other feature selection methods selected for comparison. Then, detailed experimental procedures using the proposed method as well as the filter methods is given in Section 3. In Section 4 we provide the evaluation results and discussion, comparing performance and time of computation vis-à-vis state-of-the-art results obtained by researchers in related studies. Finally, the conclusion and future research direction are presented in Section 5.

2. Feature Selection

Feature selection is widely used in many domains: intrusion detection,[25,26] genomic analysis,[27,28] text categorization,[29] and bioinformatics,[30] among others. As this work is an extension of our previous work,[31] a thorough review of the application of feature selection in intrusion detection can be found in the previous work. In this study, emphasis is given on the effectiveness and efficiency of the proposed feature selection approach in comparison to various feature selection methods.

Selecting the most useful and relevant features in a large dataset is an important means of reducing computational complexity and increasing the efficiency of models. Feature Selection (FS) is one of the successful pre-processing techniques for selecting an optimal relevant subset of features from original features. Feature selection algorithms can be broadly classified as follows:[32]

I. *Filter method*: relies on the general characteristics of the data to evaluate and select feature subsets. It separates feature selection from classifier learning so that the bias of a learning algorithm does not interact with the bias of a feature selection algorithm.

II. *Wrapper method*: uses the predictive accuracy of a predetermined learning algorithm to determine the quality of selected features. Improves performance but in comparison to the filter method, it is computationally expensive to run for data with a large number of features.

III. *Embedded method*: attempts to take advantage of the two methods by exploiting their different evaluation criteria in different search stages. It usually achieves comparable accuracy to the wrapper and comparable efficiency to the filter method. It first incorporates the statistical criteria, as the filter method does, to select several candidate feature subsets with a given cardinality, and then it chooses the subset with the highest classification accuracy as the wrapper method does. The embedded method performs both feature selection and model training simultaneously.

2.1 Proposed FS for IDS

The accuracy of an IDS model can be affected by an irrelevant and redundant feature that the intrusion detection datasets inevitably contain;[33] to reduce their effects, many researchers turned to feature selection algorithms to select only the important features.[34] In this work, we propose a wrapper-based feature selection approach with a decision tree algorithm as the feature evaluator to select optimal features and remove the redundant and irrelevant features. Our proposal is based on the following reasons:

I. Most of the existing IDS datasets contain categorical features [35] and a decision tree can handle both categorical and numeric features.[36]

II. Decision tree is a low-bias algorithm;[37] thus, it can select optimal features while avoiding underfitting, which is one of the challenging issues in classification tasks.[38]

III. Decision tree can be used to implement a trade-off between the performance of the selected features and the computation time which is required to find a good subset of features.[39] Thus, it can be stopped at any time, providing sub-optimal feature subsets.

2.2 FS Methods as a Benchmark for Comparison

To assess the effectiveness of our proposed method, three filter selection methods are used for comparisons with the full-featured UNSW-NB15 dataset models as the baseline. A comparison with state-of-the-art results is also performed. Table 1 summarises the four feature selection methods. The explanation of our proposed method is given in the methodology section, a brief explanation of the three filter-based FS methods and their basic framework is provided below.

2.2.1 Information Gain (Info Gain or IG)

This is one of the most common feature evaluation techniques. IG evaluate the worth of a feature by measuring the expected reduction in information entropy with respect to the class.[40] The formula of the information gain is shown below:

$$\mathbf{Info\ Gain}\ (\mathbf{Class}, \mathbf{Feature}) = \mathbf{C}(\mathbf{Class}) - \mathbf{C}(\mathbf{Class} \mid \mathbf{Feature}) \tag{1}$$

where C is the change in information entropy.

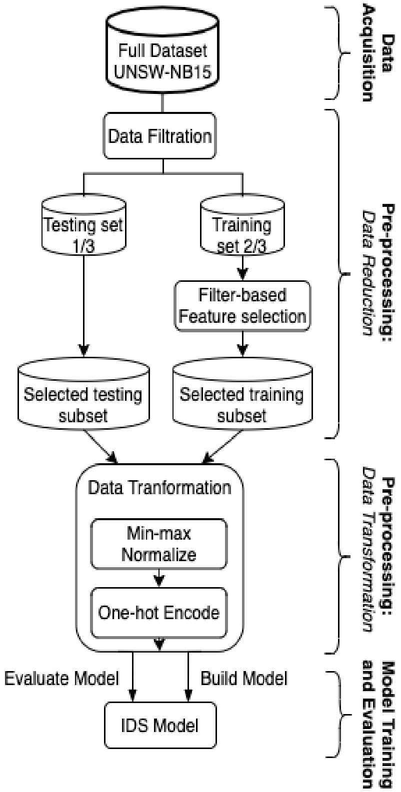

Figure 1: Filter-based FS and Model Training.

2.2.2 Gain Ratio (GR)

The Info Gain favors features with many values. The gain ratio seeks to avoid this bias by incorporating another term, split information, that is sensitive to how broadly and uniformly the considered data is split.[41] The gain ratio is defined as:

$$\mathbf{Gain\ Ratio}\ (\mathbf{Class}, \mathbf{Feature}) = \left(\mathbf{C}(\mathbf{Class}) - \mathbf{C}(\mathbf{Class} \mid \mathbf{Feature})\right) / \ \mathbf{C}(\mathbf{Feature}) \quad (2)$$

2.2.3 Relief Filter

This method evaluates the worth of a feature by repeatedly sampling an instance and considering the value of the given feature for the nearest instance of the same and different classes. In other words, Relief estimates the quality of features according to how well their values distinguish between instances that are near each other. It can operate on both discrete and continuous class data.[42]

Table 1. Summary of the FS Methods.

Name	FS Method	Feature Evaluator	Search Method
Decision Tree-Based	Wrapper	WrapperSubsetEval with J48 decision tree as a classifier	Bestfirst, forward
Information Gain	Filter	InfoGainAttributeEval	Ranker
Gain Ratio	Filter	GainRatioAttributeEval	Ranker
Relief Filter	Filter	ReliefAttributeEval	Ranker

3. Methodology

The experiment is conducted in four basic machine learning steps (i.e. data acquisition, data pre-processing, model selection and training, and model evaluation) using some experimental tools as explained below.

3.1 Experimental Tools

In literature, many tools were used for implementing, evaluating, and comparing various IDS works. WEKA, general-purpose programming languages, and Matlab were the most used tools.[43] WEKA and Python, in addition to Excel, are used in this work for data analysis and exploration, pre-processing, implementing, and validating the IDS models. Jupyter Notebook is the execution environment used for Python and its libraries.

3.2 Data Acquisition

The UNSW-NB15 dataset is used in this work. It is among the latest and recommended datasets for benchmarking [44] and is found to be reliable, good for modern-day IDS modelling.[45]

3.2.1 UNSW-NB15 Dataset

The UNSW-NB15 dataset is a new IDS dataset created at the Australian Centre for Cyber Security (ACCS) in 2015. About 2.5 million samples or 100GB of raw data were captured in modern network traffic including normal and attack behaviours and are simulated using the IXIA Perfect Storm tool and a tcpdump tool. 49 features were created using the Argus tool, the Bro-IDS tool, and 12 developed algorithms. The created features can be categorized into five groups: flow features, basic features, content features, time features, and additional generated features. The dataset has nine different modern attack types: Backdoor, DoS, Generic, Reconnaissance, Analysis, Fuzzers, Exploit, Shellcode, and Worms.[46] The UNSW-NB15 is considered as a new benchmark dataset that can be used for IDSs evaluation by the NIDS research community [47] and is recommended by.[48] For easy use and work reproducibility, the UNSW-NB15 comes along with predefined splits of a training set (175,341 samples) and a testing set (82,332 samples),[49] the predefined training and testing sets are used in this work. The publicly available training and testing set both contain only 44 features: 42 attributes and 2 classes. Since our primary focus is binary classification, the broad distribution of total attacks (anomaly) and normal traffic samples of the training and testing sets are used as shown in Table 2.

Table 2. UNSW-NB15 Distribution Sample.

Category	Training Set		Testing Set	
	Size	**Distribution (%)**	**Size**	**Distribution (%)**
Total Attacks	119,341	68.06	45,332	55.06
Normal	56,000	31.94	37,000	44.94
Overall Samples	**175,341**	**100**	**82,332**	**100**

3.3 Data Pre-processing

Two major pre-processing steps were performed, namely, data reduction (filtration and feature selection) and data transformation (data normalization and encoding).

3.3.1 Data Reduction

3.3.1.1 Data Filtration

The UNSW-NB15 dataset comes with 42 attributes, 2 class attributes, and an additional id attribute that is removed; some irrelevant data in both the training and testing set are removed. And since we are only interested in binary classification, the class attribute *attack_cat* indicating the categories of attacks and normal traffic is removed before feature selection.

3.3.1.2 Feature Selection

To avoiding information leakage and subsequent building of misleading or overfitting models,[50] only the training set is used in feature selection. The testing set is solely used to assess the performance of the models.

We propose a wrapper-based DT approach where the BestFirst Forward search strategy is used in feature search with five consecutive non-improving nodes as the search stopping criteria and accuracy as the evaluation measure. For the feature evaluator, J48 – a java implementation of Quinlan's C4.5,[51] decision tree algorithm [52] available in WEKA [53] is used. A total of 19 optimal features are selected by the wrapper-based FS approach, and Figure 2 below depicts the entire wrapper feature selection and modelling process. For the filter methods, the default WEKA evaluator and Ranker search method setup of each filter is used. Since the filter methods rank all the features by their evaluations, the 19 top-ranked features are selected in each, and Figure 1 above depicts the filter feature selection and their modelling process. After the feature selection operations, a supervised attribute Remove filter in WEKA is used to collect the features subsets in all the feature selection methods. Table 3 shows the selected features.

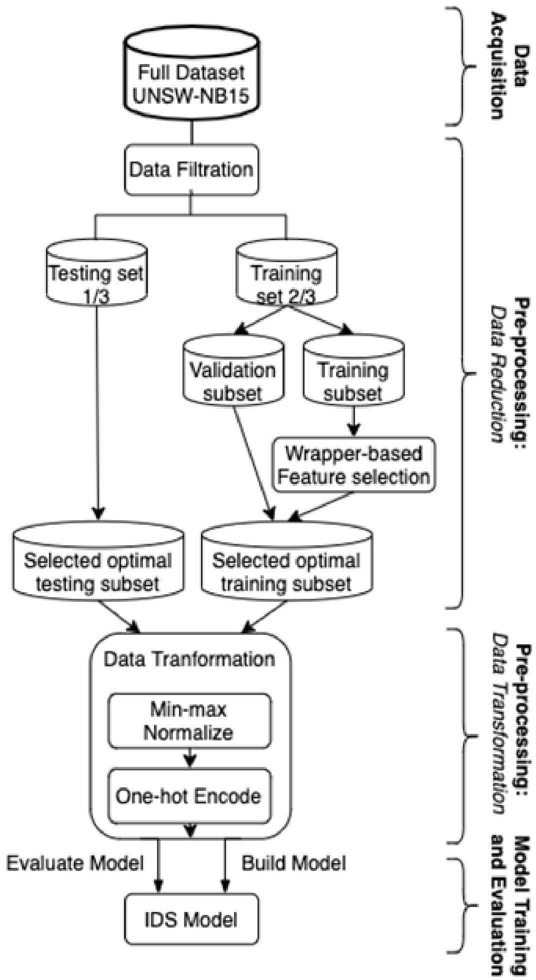

Figure 2: Wrapper-based FS and Model Training.

3.3.2 Data Transformation

3.3.2.1 Data Normalization

Numeric and categorical/nominal features are the two types of features in the UNSW-NB15 dataset. To avoid classifier bias towards numeric features with large value ranges, min-max normalization with a range of 0 to 1 is applied on all the numeric features across the datasets using Equation (3) below. To avoid affecting the feature selection process, the normalization process is performed after the feature selection.

$$x_{new} = \frac{x - min(x)}{max(x) - min(x)} \qquad (3)$$

Table 3. Selected Features.

FS Method	FS No.	Selected Features
DT-Based Wrapper	19	*proto, *service, spkts, sbytes, dbytes, dttl, sloss, dloss, swin, stcpb, trans_depth, response_body_len, ct_srv_src, ct_src_dport_ltm, ct_dst_sport_ltm, ct_dst_src_ltm, ct_flw_http_mthd, ct_src_ltm, ct_srv_dst
Info Gain Filter	19	sbytes, dbytes, sttl, dttl, ct_state_ttl, rate, sload, smean, dur, dmean, dinpkt, dpkts, dload, sinpkt, tcprtt, synack, ackdat, sjit, spkts
Gain Ration Filter	19	sttl, dttl, ct_state_ttl, is_sm_ips_ports, *state, ackdat, tcprtt, synack, dinpkt, dload, dbytes, dpkts, rate, sbytes, dmean, dur, ct_dst_sport_ltm, response_body_len, smean
Relief Filter	19	*service, *proto, dttl, sttl, ct_dst_sport_ltm, smean, ct_state_ttl, ct_dst_ltm, ct_src_ltm, ct_src_dport_ltm, dload, ct_srv_dst, ct_srv_src, rate, ct_dst_src_ltm, dmean, is_sm_ips_ports, dtcpb, stcpb
(*) – indicates nominal features		

3.3.2.2 Data Encoding

All the nominal features are one-hot encoded. The full UNSW-NB15 dataset has 39 numeric and 3 nominal features, the nominal features are proto, service, and state. All the 19 features selected by the Information Gain filter are numeric. The Gain Ration filter selects 18 numeric and only one nominal feature (*state*). The DT-based wrapper and the Relief filter selected two nominal features (*proto*, and *service*). An example of a one-hot encoding of *protocol_type* feature with three sample values is shown in Table 4. Because one-hot encoding increases the dataset dimension, so to avoid losing some nominal features' values encoded during feature selection, the encoding is performed after the feature selection and normalization processes.

Table 4. One-Hot Encoding Example.

Protocol_type
UDP
TCP
ICMP

UDP	TCP	ICMP
1	0	0
0	1	0
0	0	1

The dimensions of the final datasets increased as shown in Table 5 after encoding the features. The final encoded features are then used in training the models.

Table 5. Final Datasets Dimensions.

Dataset	UNSW-NB15 Dataset features	
	Before Encoding	After Encoding
Full dataset	42	194
DT Wrapper	19	163
IG Filter	19	19
GR Filter	19	27
Relief Filter	19	163

3.4 Model Selection and Training

Building the models constitutes of two stages: the training stage and the testing stage. So, the dataset is divided into two sets, the training set and the testing set using the hold-out method. During the training stage, the algorithms are trained using the training set, then in the testing stage, the testing set is used to assess the performance and reliability of the built IDS models. Figure 1 and Figure 2 depicted the entire model training and testing processes. Using the full and the various FS datasets, the selected algorithms are used to build a total of 25 models. To measure the effectiveness of our FS methods, some evaluation metrics are used to evaluate and compare the models. The model evaluation metrics and the result of the evaluations are provided in the next subsection and Result and Discussion section of this work respectively.

3.5 Model Evaluation Metrics

Classification accuracy, detection rate (DR), and false alarm rate (FAR) are the most used metrics in IDS works.[54] In this work, these metrics are adopted in addition to computational time. The formulas associated with the metrics are as follow:

$$\boldsymbol{Accuracy(ACC)} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}} \tag{4}$$

$$\boldsymbol{Detection\ Rate(DR)} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \tag{5}$$

$$\boldsymbol{False\ Alert\ Rate(FAR)} = \frac{\boldsymbol{FP}}{\boldsymbol{FP+TN}} \tag{6}$$

The computational time is the entire time taken to train and evaluate a model, including the FS time. Because the timing depends on factors beyond our control such as CPU task switching, etc., we avoided running heavy tasks whilst building the models, we also try preventing the computer from sleeping to ensure minimal interference.

4. Result and Discussion

The experimental platform, the result, and its interpretations are presented in this section. Comparations of the models built using the different feature selection approach as well as with state-of-the-art IDS works are made.

4.1 Experimental Platform

All the models are implemented and executed in the same environment using the same programming language as shown in Table 6.

Table 6. Experimental Platform.

Name	Details
Computer	Lenovo ThinkPad T450
OS	Windows 7 Ultimate 64-bit
CPU	2.30GHz Intel Core i5 series 5 processor
RAM	8GB (7.70GB usable)
Storage Disk	240GB SSD
Execution platform	Jupyter Notebook
Experimental Tools	Excel, WEKA, Python

4.2 Performance and Computational Time Comparisons

Five models are built with each of the selected algorithms. In this sub-section, comparisons of models built using our proposed method and those built using the three filter-based feature selection methods are made with the models built using the full features of the UNSW-NB15 dataset as the baseline models. The basis of the comparisons is the performance and the computation time shown in Table 7 and Table 8 respectively.

In ANN models, our methods perform rather poorly achieving the third-best score on accuracy and FAR with the third-highest computation time. It improves computation time but not the performance in comparison to the baseline model. In comparison to the baseline and the filter-based methods, our method achieves the worst performance on SVM across all metrics with the highest overall computational time of all models. Against the baseline model, our method improves neither performance nor computational time, making it the SVM the worst of the five models. Our method also performed poorly on the KNN model achieving the worst on accuracy and FAR as well as third-best on DR and computational time against all other KNN models. Thus, it improves on computational time but not on performance scores.

The proposed wrapper-based method achieved its best performance on the RF model with an accuracy of 86.41 %, which is the third-best, after baseline and relief filter-based models. It achieves the best FAR with third-best computation time with two of the filter-based models taking less computation time. Our method failed to improve model performance against the baseline but it does improve on computational time. With NB models, our method achieves similar performance to the baseline model in lower time, and against other methods, though our method achieves joint best FAR on NB models, it has the second-lowest detection rate (DR), which is more important than the other metrics in IDS.[55] In terms of computational time, our method has the third-best, after IG and GR filter-based models which also achieves worst on FAR than our method.

Table 7. Models Performance Comparisons.

Models	Evaluation metrics	UNSW-NB15				
		Full Features	DT Wrapper	IG Filter	GR Filter	Relief Filter
ANN	ACC	86.00	82.08	82.06	83.72	**86.51**
	DR	98.62	97.94	99.41	98.39	**97.99**
	FAR	29.45	37.36	39.19	34.26	**27.56**
SVM	ACC	**81.6**	79.11	80.87	80.92	81.58
	DR	**99.64**	99.31	99.84	99.87	99.60
	FAR	**40.51**	45.64	42.38	42.29	40.50
KNN	ACC	84.78	83.21	86.1	**86.96**	84.81
	DR	96.46	96.44	96.01	**95.90**	96.52
	FAR	29.53	33.01	26.04	**23.98**	29.53
RF	ACC	**86.82**	*86.41*	86.14	85.98	86.49
	DR	**98.7**	*97.95*	97.8	98.00	98.79
	FAR	**27.74**	*27.73*	28.16	28.75	28.58
NB	ACC	55.61	55.61	76.37	71.28	55.44
	DR	19.39	19.38	93.7	98.16	19.07
	FAR	0.01	0.01	44.86	61.65	0.01

Overall, both the performance and computation time of models built using our method in comparison to the baseline and filter-based models are not satisfactory. Wrapper feature selection method is mainly known for its robustness in selecting the best possible features to improve performance at the cost of a computation time,[56] with our proposed method, however, there is not any significant performance improvement despite the huge among of time taken in the feature selection process and in training the models which, on average, is higher than that of all the other methods. Thus, it can be deduced that, in the case of IDS modelling with UNSW-NB15 dataset, filter-based feature selection methods, like the ones used in this work, are better and should be considered as they performed fairly equivalent to our proposed wrapper method on corresponding models and also, they take less time as can be seen in Table 8 and Figure 3 below. Different wrapper feature evaluators, search approaches, and/or termination conditions can also be considered.

Table 8. Models Computation Time Comparisons.

Dataset Features	FS Time	Models Training Time					Overall Training Time Average
		ANN	SVM	KNN	RF	NB	
Full Features	N/A	11.27m	181.68m	17.92m	0.74m	4.64s	42.34m
DT Wrapper	31.4hrs	4.95m	259.1m	10.94m	0.63m	2.86s	**55.13m**
IG Filter	12.00s	4.85m	65.86m	1.82m	0.54m	1.14s	14.62m
GR Filter	12.00s	2.41m	111.12m	3.37m	0.46m	1.4s	23.48m
Relief Filter	3.92hrs	8.00m	193.65m	16.79m	0.65m	3.08s	43.83m

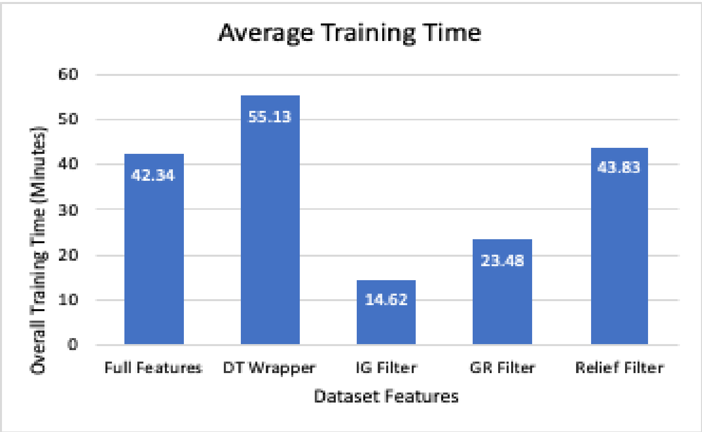

Figure 3. Models' Average Training Time.

Furthermore, although the performance and computation time of some of the used algorithms can be influenced by other factors such as normalization,[57] feature selection normally improves both the performance and computation time of algorithms.[58] However, as seen in the comparisons made, these expectations failed to occur on many models with no significant improvement spotted on individual models against the baseline models. Thus, this essentially raises some concerns about the conformant nature of the UNSW-NB15 dataset.

4.3 Comparisons with Other works

To assess the effectiveness of our proposed method, we selected the best performing model, from among the models implemented using the method for corresponding comparisons with other state-of-the-art IDS works. There are many similar research works, we however limited our comparison to those that also used feature selection on the UNSW-NB15 dataset. We compare the percentages of accuracy (ACC), attack detection rate (DR), and false alert rate (FAR) whilst also paying attention to feature selection method, number of features, and algorithms used. Table 9 shows the performance comparisons of the works chronologically.

Table 9. Comparison with Related Works.

[Work] Year	FS Method	FS no.	Algorithm	ACC (%)	DR (%)	FAR (%)
[59] 2015	ARM-Based	11	LR	83.0	68	14.2
[60] 2017	GA-LR	20	DT	81.42	–	6.39
[61] 2017	Functional measures	33	DL-binomial	98.99	95.84	**0.56**
[62] 2019	DSAE	10	DL-Soft-max	89.13	–	0.75
[63] 2019	K-means	41	DNN	**99.19**	–	–
[64] 2020	NSGAII-ANN	19	RF	94.8	94.8	6.0
This Work	*DT-based*	*19*	*RF*	*86.41*	***97.95***	*27.73*

From Table 9 it can be seen that our method achieved the best DR of 97.95 %, performed better than two of the works in ACC, and has the height FAR. The best ACC and FAR are achieved by [65] and [66] respectively, both of which used deep learning classifiers. Their good results may be influenced by the use of deep learning classifiers which, recently, are proving to be good in IDS classification tasks,[67,68,69] It is important to note that in IDS, not detecting an attack can be costlier than mis-detecting an attack [70], thus DR can be more important than any other reported metrics, and hence, our method can be more effective in detecting an attack than both [71] and [72]. Overall, our proposed method is quite effective. Its major downside is the expensive computational time required, however, giving that IDSs are kind of systems that can be trained offline and deployed online for use,[73] this would not have been a major point of concern had our method improve performance against the baseline models and also achieved better performance in comparison to filter-based methods.

5. Conclusion

In this work, we analysed wrapper-based and filter-based modelling approaches. Various IDS models are built and their performance and accuracy were evaluated. The models built using filter methods achieved results similar to that of the models built using wrapper methods at considerably lower feature selection and model training computation time. The wrapper feature selection method is generally expected to improve performance, however, it failed to do so; instead, it greatly increases the computational time. Thus, although the wrapper method is rated good in comparison to state-of-the-art works, utilizing it in IDS modelling, especially while working with the UNSW-NB15 dataset, might not produce most effective results. However, the use of different wrapper-based feature selection procedures or filter-based feature selection methods such as the ones used in this work in IDS modelling using the UNSW-NB15 dataset is recommended. Finally, our work also highlighted the need for a more in-depth analysis of the conformity of the UNSW-NB15 dataset.

Some interesting and important future works can be performed particularly on reducing the high false alert rate observed because, besides a high detection rate, a good IDS should have a very low false alert rate. Furthermore, this work primarily focused on binary classification, however like most of the IDS datasets, the used dataset contained various attack types, thus multi-classification work can be performed. Finally, more recent datasets such as the IDS 2017 and IDS 2018 that have been widely used for benchmarking, can also be utilized.

References

1 Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu, "A Survey of Network Anomaly Detection Techniques," *Journal of Network and Computer Applications* 60 (2016): 19-31, https://doi.org/10.1016/j.jnca.2015.11.016.

2 Rung Ching Chen and Su Ping Chen, "An Intrusion Detection Based on Support Vector Machines with a Voting Weight Schema," *Lecture Notes in Computer Science* 4570 LNAI (2007): 1148–57, https://doi.org/10.1007/978-3-540-73325-6_115.

3 Wei-Chao Lin, Shih-Wen Ke, and Chih-Fong Tsai, "CANN: An Intrusion Detection System Based on Combining Cluster Centers and Nearest Neighbors," *Knowledge-Based Systems* 78, no. 1 (2015): 13-21, https://doi.org/10.1016/j.knosys.2015.01.009.

4 Tom M. Mitchell, *Machine Learning*. *Intelligent Systems Reference Library* (McGraw-Hill, 1997).

5 Atilla Özgür and Hamit Erdem, "A Review of KDD99 Dataset Usage in Intrusion Detection and Machine Learning between 2010 and 2015," *PeerJ Preprints* 4 (2016): 1-21. https://doi.org/10.7287/peerj.preprints.1954.

6 Yuyang Zhou, Guang Cheng, Shanqing Jiang, and Mian Dai, "Building an Efficient Intrusion Detection System Based on Feature Selection and Ensemble Classifier," *Computer Networks* 174 (2020), https://doi.org/10.1016/j.comnet.2020.107247.

7 Neha Acharya and Shailendra Singh, "An IWD-Based Feature Selection Method for Intrusion Detection System," *Soft Computing* 22, no. 13 (2018): 4407-16, https://doi.org/10.1007/s00500-017-2635-2.

8 Markus Ring, Sarah Wunderlich, Deniz Scheuring, Dieter Landes, and Andreas Hotho, "A Survey of Network-Based Intrusion Detection Data Sets," *Computers and Security* 86 (2019): 147-167, https://doi.org/10.`1016/j.cose.2019.06.005.

9 Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu, "Feature Selection: A Data Perspective," *ACM Computing Surveys* 50, no. 6 (2017): 128-130, https://doi.org/https://doi.org/10.1145/3136625.

10 Shadi Aljawarneh, Monther Aldwairi, and Muneer Bani Yassein, "Anomaly-Based Intrusion Detection System through Feature Selection Analysis and Building Hybrid Efficient Model," *Journal of Computational Science* 25 (2018): 152-160, https://doi.org/10.1016/j.jocs.2017.03.006.

11 Wei Wang, Xiangliang Zhang, Sylvain Gombault, and Svein J. Knapskog, "Attribute Normalization in Network Intrusion Detection," in *10th International Symposium on Pervasive Systems, Algorithms, and Networks*, I-SPAN 2009, 448–53, IEEE Computer

Society, 2009, https://doi.org/10.1109/I-SPAN.2009.49.

12 Weijun Li and Zhenyu Liu, "A Method of SVM with Normalization in Intrusion Detection," *Procedia Environmental Sciences* 11 (2011): 256-262, https://doi.org/10.1016/j.proenv.2011.12.040.

13 Zhou, Cheng, Jiang, and Dai, "Building an Efficient Intrusion Detection".

14 Sumaiya Thaseen and Ch Aswani Kumar, "An Analysis of Supervised Tree Based Classifiers for Intrusion Detection System," *Proceedings of the 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering, PRIME 2013*, 294-299, Salem, India, 2013, https://doi.org/10.1109/ICPRIME.2013.6496489.

15 Acharya and Singh, " An IWD-Based Feature Selection Method."

16 Tarrah R. Glass-Vanderlan, Michael D. Iannacone, Maria S. Vincent, Qian Chen, and Robert A. Bridges, "A Survey of Intrusion Detection Systems Leveraging Host Data," *ACM Computing Surveys* 52, no. 6 (2018): 1-40, https://doi.org/10.1145/3344382.

17 Jiliang Tang, Salem Alelyani, and Huan Liu, "Feature Selection for Classification: A Review," in *Data Classification: Algorithms and Applications,* ed. Charu C. Aggarwal (New York: Chapman and Hall/CRC, 2014), https://doi.org/10.1201/b17320.

18 Tang, Alelyani, and Liu, "Feature Selection for Classification: A Review."

19 Mubarak Albarka Umar, Chen Zhanfang, and Yan Liu, "Network Intrusion Detection Using Wrapper-Based Decision Tree for Feature Selection," in *ICICSE '20: International Conference on Internet Computing for Science and Engineering*, 5–13, Association for Computing Machinery Digital Library, 2020, https://doi.org/10.1145/3424311.3424330.

20 Özgür and Erdem, "A Review of KDD99."

21 Ring, et al., "A Survey of Network-Based Intrusion Detection Data Sets."

22 Mubarak Albarka Umar and Chen Zhanfang, "Effects of Feature Selection and Normalization on Network Intrusion Detection," *TechRxiv*, Preprint, 2020, https://doi.org/10.36227/techrxiv.12480425.

23 Nour Moustafa and Jill Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set)," in *Proceedings of 2015 Military Communications and Information Systems Conference, MilCIS 2015,* Canberra, ACT, Australia, 2015, https://doi.org/10.1109/MilCIS.2015.7348942.

24 Özgür and Erdem, "A Review of KDD99."

25 Mohammed A. Ambusaidi, Xiangjian He, Zhiyuan Tan, Priyadarsi Nanda, Liang Fu Lu, and Upasana T. Nagar, "A Novel Feature Selection Approach for Intrusion Detection Data Classification," in *Proceedings IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2014*, 82–89, Beijing, China, 2015, https://doi.org/10.1109/TrustCom.2014.15.

26 Seung Ho Kang and Kuinam J. Kim, "A Feature Selection Approach to Find Optimal Feature Subsets for the Network Intrusion Detection System," *Cluster Computing* 19, no. 1 (2016): 325-333, https://doi.org/10.1007/s10586-015-0527-8.

27 Iñaki Inza, Pedro Larrañaga, Rosa Blanco, and Antonio J. Cerrolaza, "Filter versus Wrapper Gene Selection Approaches in DNA Microarray Domains," *Artificial Intelligence in Medicine* 31, no. 2 (2004): 91–103, https://doi.org/10.1016/j.artmed.2004.01.007.

28 Hicham Omara, Mohamed Lazaar, and Youness Tabii, "Effect of Feature Selection on

Gene Expression Datasets Classification Accuracy," *International Journal of Electrical and Computer Engineering* 8, no. 5 (2018): 3194-3203, https://doi.org/10.11591/ijece.v8i5.pp3194-3203.

[29] George Forman, "An Extensive Empirical Study of Feature Selection Metrics for Text Classification," *Journal of Machine Learning Research* 3 (2003): 1289-1305.

[30] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga, "A Review of Feature Selection Techniques in Bioinformatics," *Bioinformatics Review* 23, no. 19 (2007): 2507-2517, https://doi.org/10.1093/bioinformatics/btm344.

[31] Umar, Zhanfang, and Liu, "Network Intrusion Detection Using Wrapper-Based Decision Tree for Feature Selection."

[32] Tang, Alelyani, and Liu, "Feature Selection for Classification: A Review."

[33] Yuyang Zhou, Guang Cheng, Shanqing Jiang, and Mian Dai, "An Efficient Intrusion Detection System Based on Feature Selection and Ensemble Classifier," *Journal of Latex Class Files* 14, no. 8 (2015): 1–12, http://arxiv.org/abs/1904.01352.

[34] You Chen, Yang Li, Xue Qi Cheng, and Li Guo, "Survey and Taxonomy of Feature Selection Algorithms in Intrusion Detection System," *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4318 (2006): 153–67, https://doi.org/10.1007/11937807_13.

[35] Ring et al., "A Survey of Network-Based Intrusion Detection Data Sets."

[36] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, *An Introduction to Statistical Learning* (New York: Springer, 2013).

[37] Gangadhar Shobha, and Shanta Rangaswamy, "Machine Learning," *Handbook of Statistics* 38 (2018): 197-228, https://doi.org/10.1016/bs.host.2018.07.004.

[38] Kenneth P. Burnham and David R. Anderson, *Model Selection and Inference: A Practical Information-Theoretic Approach* (New York: Springer-Verlag Inc., 2002), https://doi.org/10.2307/3803117.

[39] Salvador García, Julián Luengo, and Francisco Herrera, *Data Preprocessing in Data Mining*, Intelligent Systems Reference Library 72 (Springer, 2015), https://doi.org/10.1007/978-3-319-10247-4_8.

[40] Richard Jensen and Qiang Shen, *Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches* (Wiley-IEEE Press, 2008), https://doi.org/10.1002/9780470377888.

[41] Jensen and Shen, *Computational Intelligence and Feature Selection.*

[42] Kenji Kira and Larry A. Rendell, "The Feature Selection Problem: Traditional Methods and a New Algorithm," in *AAAI'92: Proceedings of the Tenth National Conference on Artificial Intelligence*, July 1992, pp. 129-134, https://dl.acm.org/doi/10.5555/1867135.1867155.

[43] Özgür and Erdem,"A Review of KDD99."

[44] Ring et al., "A Survey of Network-Based Intrusion Detection Data Sets."

[45] Umar and Zhanfang, "Effects of Feature Selection and Normalization on Network Intrusion Detection."

[46] Moustafa and Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems."

[47] Nour Moustafa and Jill Slay, "The Evaluation of Network Anomaly Detection Systems:

Statistical Analysis of the UNSW-NB15 Data Set and the Comparison with the KDD99 Data Set," *Information Security Journal* 25, no. 1-3 (2016): 18-31, https://doi.org/10.1080/19393555.2015.1125974.

48 Ring et al., "A Survey of Network-Based Intrusion Detection Data Sets."

49 UNSW Sydney, "The UNSW-NB15 Data Set Description," 2015, https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/.

50 Ian H. Witten, Eibe Frank, and Mark A. Hall, *Data Mining Practical Machine Learning Tools and Techniques*, 3rd edition (Morgan Kaufmann Publishers, 2011).

51 Steven L. Salzberg, "C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993," *Machine Learning* 16 (1994): 235-240, https://doi.org/10.1007/BF00993309.

52 Witten, Frank, and Hall, *Data Mining Practical Machine Learning Tools and Techniques*.

53 Geoffrey Holmes, Andrew Donkin, and Ian H Witten, "WEKA: A Machine Learning Workbench," In *Proceedings of ANZIIS '94 - Australian New Zealand Intelligent Information Systems Conference,* Brisbane QLD, Australia, IEEE Xplore, 06 August 2002, https://doi.org/10.1109/ANZIIS.1994.396988.

54 Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman, "Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges," *Cybersecurity* 2, no. 1 (2019), article # 20, https://doi.org/10.1186/s42400-019-0038-7.

55 Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and Lofti Zadeh, *Feature Extraction: Foundations and Application* (Berlin, Heidelberg: Springer, 2006).

56 Tang, Alelyani, and Liu, "Feature Selection for Classification: A Review"; Huan Liu and Lei Yu, "Toward Integrating Feature Selection Algorithms for Classification and Clustering," *IEEE Transactions on Knowledge and Data Engineering* 17, no. 4 (2005): 491-502, https://doi.org/10.1109/TKDE.2005.66.

57 Xindong Wu, Vipin Kumar, Quinlan J. Ross, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, et al., "Top 10 Algorithms in Data Mining," *Knowledge and Information Systems* 14, no. 1 (2008): 1–37, https://doi.org/10.1007/s10115-007-0114-2.

58 Tang, Alelyani, and Liu, "Feature Selection for Classification: A Review."

59 Nour Moustafa and Jill Slay, "A Hybrid Feature Selection for Network Intrusion Detection Systems: Central Points and Association Rules," In *16th Australian Information Warfare Conference*, 5–13, 2015, https://doi.org/10.4225/75/57a84d4fbefbb.

60 Chaouki Khammassi and Saoussen Krichen, "A GA-LR Wrapper Approach for Feature Selection in Network Intrusion Detection," *Computers and Security* 70 (2017): 255-277, https://doi.org/10.1016/j.cose.2017.06.005.

61 Malek Al-Zewairi, Sufyan Almajali, and Arafat Awajan, "Experimental Evaluation of a Multi-Layer Feed-Forward Artificial Neural Network Classifier for Network Intrusion Detection System," in *Proceedings - 2017 International Conference on New Trends in Computing Sciences (ICTCS)*, 167-172, Amman, Jordan, 2017, https://doi.org/10.1109/ICTCS.2017.29.

62 Farrukh Aslam Khan, Abdu Gumaei, Abdelouahid Derhab, and Amir Hussain, "A Two-Stage Deep Learning Model for Efficient Network Intrusion Detection," *IEEE Access* 7 (2019): 30373-85, https://doi.org/10.1109/ACCESS.2019.2899721.

[63] Osama Faker and Erdogan Dogdu, “Intrusion Detection Using Big Data and Deep Learning Techniques,” in *ACMSE 2019 - Proceedings of the 2019 ACM Southeast Conference*, 86-93, 2019, https://doi.org/10.1145/3299815.3314439.

[64] Anahita Golrang, Alale Mohammadi Golrang, Sule Yildirim Yayilgan, and Ogerta Elezaj, “A Novel Hybrid IDS Based on Modified NSGAII-ANN,” *Electronics* 9, no. 4 (2020), article # 577, https://doi.org/10.3390/electronics9040577.

[65] Faker and Dogdu, “Intrusion Detection Using Big Data and Deep Learning.”

[66] Al-Zewairi, Almajali, and Awajan, “Experimental Evaluation of a Multi-Layer Feed-Forward Artificial Neural Network Classifier.”

[67] Bojan Kolosnjaji, Apostolis Zarras, George Webster, and Claudia Eckert, “Deep Learning for Classification of Malware System Call Sequences,” *Lecture Notes in Computer Science* (2016): 137-149, https://doi.org/10.1007/978-3-319-50127-7_11.

[68] Ahmad Javaid, Quamar Niyaz, Weiqing Sun, and Mansoor Alam, “A Deep Learning Approach for Network Intrusion Detection System,” *EAI Endorsed Transactions on Security and Safety* 16, no. 9 (2016): e2, https://doi.org/10.4108/eai.3-12-2015.2262516.

[69] R. Vinayakumar, K. P. Soman, Prabaharan Poornachandran, and S. Akarsh, “Application of Deep Learning Architectures for Cyber Security,” in *Cybersecurity and Secure Information Systems. Advanced Sciences and Technologies for Security Applications*, eds. Ella Hassanien and Mohamed Elhoseny (Springer, Cham, 2019), 125-160, https://doi.org/10.1007/978-3-030-16837-7_7.

[70] Guyon, Gunn, Nikravesh, and Zadeh, *Feature Extraction: Foundations and Application.*

[71] Faker and Dogdu, “Intrusion Detection Using Big Data and Deep Learning.”

[72] Al-Zewairi, Almajali, and Awajan, “Experimental Evaluation of a Multi-Layer Feed-Forward Artificial Neural Network Classifier.”

[73] Monowar H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, “Network Anomaly Detection: Methods, Systems and Tools,” *IEEE Communications Surveys and Tutorials* 16, no. 1 (2014): 303–36, https://doi.org/10.1109/SURV.2013.052213.00046.

About the Authors

Mubarak Albarka **Umar**, BSc, Software Engineering, University of East London (2015), MEng, Computer Applied Technology, Changchun University of Science and Technology (2020). He is a staff member of Katsina State Institute of Technology and Management (KSITM), Katsina, Nigeria. He has published eight articles in international journals. His research interests include data mining, soft computing, software engineering and network security.

Zhanfang **Chen**, Ph.D., Associate Professor of School of Computer Science and Technology in Changchun University of Science and Technology, China. His research interests include network engineering, computer architecture, data mining, soft computing, and software engineering. He is currently a lecturer at Changchun University of Science and Technology. With over 15 years of experience, Dr. Chen has worked on over 20 research projects and has published over 20 articles in international journals.

Yan **Liu**, Ph.D., Senior Lecturer of School of Computer Science and Technology in Changchun University of Science and Technology, China. Her research interests include supply chain management, software engineering, information systems (business informatics), and expert systems. She is currently a lecturer at Changchun University of Science and Technology. She has published over eight articles in international journals.