Hindawi
Security and Communication Networks
Volume 2020, Article ID 4586875, 9 pages
https://doi.org/10.1155/2020/4586875

Research Article

A Stacking Ensemble for Network Intrusion Detection Using Heterogeneous Datasets

Smitha Rajagopal, Poornima Panduranga Kundapur, and Katiganere Siddaramappa Hareesha

Department of Computer Applications, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India

Correspondence should be addressed to Katiganere Siddaramappa Hareesha; hareesh.ks@manipal.edu

Received 24 August 2019; Accepted 26 December 2019; Published 24 January 2020

Academic Editor: Stelvio Cimato

The problem of network intrusion detection poses innumerable challenges to the research community, industry, and commercial sectors. Moreover, the persistent attacks occurring on the cyber-threat landscape compel researchers to devise robust approaches in order to address the recurring problem. Given the presence of massive network traffic, conventional machine learning algorithms when applied in the field of network intrusion detection are quite ineffective. Instead, a hybrid multimodel solution when sought improves performance thereby producing reliable predictions. Therefore, this article presents an ensemble model using metaclassification approach enabled by stacked generalization. Two contemporary as well as heterogeneous datasets, namely, UNSW NB-15, a packet-based dataset, and UGR'16, a flow-based dataset, that were captured in emulated as well as real network traffic environment, respectively, were used for experimentation. Empirical results indicate that the proposed stacking ensemble is capable of generating superior predictions with respect to a real-time dataset (97% accuracy) than an emulated one (94% accuracy).

1. Introduction

Network intrusion detection is a significant research area since cyber attacks are increasing at an alarming rate [1]. Numerous studies have been put forward in order to propose noteworthy approaches for combating malicious cyber activities. However, as and when cyber attacks become more complex, the existing approaches fail to address the problem effectively. Traditional defensive strategies like firewalls, antivirus, and authentication seem to be inefficient for many complex threats because cyber-attack vectors are highly sophisticated [2]. Network intrusion detection is a major decision-making problem that can be addressed by the application of classification algorithms [3]. Several machine learning algorithms like fuzzy logic, neural networks, support vector machine, Naïve Bayes, K nearest neighbor, and decision trees have been employed in the field of network intrusion detection [4]. Whenever a combination or an ensemble approach is introduced, performance of individual algorithms can be enhanced. Ensemble paradigm is a notable machine learning approach wherein different algorithms are employed to improve predictions. Some studies have also demonstrated that the application of ensemble paradigm can prove to be versatile and certainly boost prediction accuracy and detection speed [5–8]. Going by the same assertion, the proposed approach emphasises the application of supervised machine learning algorithms to propose a classification framework using a concept called stacked generalization. As illustrated in [9–11], stacking or stacked generalization is advantageous since the concept is based on combining predictions from different individual classifiers that can substantially improve generalization too.

The advantage of stacking was explained in [12] to perform protein classification, and desirable accuracy was accomplished. As explained in [13], classifier ensembles or combiners or committees offer better solutions by handling bias-variance trade-off more effectively than individual classifiers. A comparative analysis was conducted to

analyse SVM's performance along with classifiers like AdaBoost, J48, random forest, BayesNet, and logistic regression. It was conspicuous that all the algorithmic combinations with SVM produced better results than individual SVM [14]. The implementation of the ensemble learning algorithm called super learner resulted in improved predictions using the MAWILab dataset [15]. One such ensemble learning paradigm is stacking that considers several machine learning algorithms, uses a metamodel to combine predictions from individual algorithms, and thereby improves overall performance. By combining the advantages of multiple algorithms, detection effect can be enhanced [16]. The stacking method was employed to detect malware on mobile devices that showed an improvement in accuracy and F measure [17].

2. Related Work

Several methods have been put forth by researchers to perform network intrusion detection using a combination of algorithms. This section presents an overview of such combinative approaches that focus on improving the overall performance. An emerging approach for intrusion detection involving an ensemble design was put forth using neutrosophic logic classifier, an extension to fuzzy logic. A genetic algorithm was used to generate rules. The aforesaid design could decrease the false alarm rate to 3.19% as compared to other approaches [18].

Support vector machine (SVM) is a well-known classifier that can classify from a limited set of samples given to it but still can optimize predictions [19]. It was demonstrated by Chen et al. [20] that SVM was superior to artificial neural networks (ANNs) in terms of detecting intrusions while experimenting with basic security module (BSM) audit data from Defence Advanced Research Projects Agency (DARPA) intrusion detection dataset. This is because ANN requires lot of training data, whereas SVM can perform better with relatively less data and can execute much faster. However, SVM is known to excel primarily with respect to binary classification, but when combined with other classifiers, SVM can yield better results for multiclass classification too.

An ensemble design involving multilayer perceptron and radial basis function demonstrated that superior performance could be attained by consolidating two individual models. Compared to individual models, the hybrid model devised by Govindarajan and Chandrasekaran [21] seemed to be more accurate. This study used a dataset developed at the University of New Mexico which consisted of both normal and abnormal traces pertaining to mail application.

An intrusion detection system was designed using a combination of SVM and K nearest neighbor (KNN). Particle swarm optimization (PSO) generated weights were used to create an ensemble design that accomplished an improvement of 0.756% in accuracy as compared to the best base expert [22].

Rangadurai Karthick et al. [23] developed an adaptive intrusion detection approach by combining hidden Markov and Naïve Bayesian models. Empirical results indicated that the aforementioned combinative approach yielded favourable results and learned the nature of traffic quite efficiently. Traces from Center of Applied Internet Data Analysis (CAIDA) and DARPA datasets were used to implement the hybrid model.

Another two-step hybrid method based on binary classification and KNN was proposed to decrease the bias, normally encountered pertaining to classwise predictions. Step 1 involved the usage of binary classifiers, and an aggregation module was employed to recognize abnormal connections, whereas in Step 2, KNN was used to classify those instances whose classes were undetermined after Step 1 [24].

A hybrid intrusion detection technique was proposed by Malik et al. [25] using binary particle swarm optimisation (BPSO) and random forest (RF) to classify probe attack patterns. BPSO, being a good search optimizer, and RF, an efficient classifier, contributed towards achieving better performance. This method was compared with eight other classifiers, and it was interesting to note that BPSO-RF combination yielded better results when compared to individual classifiers.

An ensemble classifier using random forest, C4.5, and forest by penalizing attributes (FPA) was proposed by Zhou and Cheng [26]. This study used average of probability (AOP) algorithm to merge the decisions from different classifiers using a modern intrusion detection dataset CIC-IDS2017. Results indicated a very good increase in accuracy, i.e., 96.76%.

An insightful study was conducted by Khammassi and Krichen [27] using a combination of genetic algorithm and decision trees, wherein the genetic algorithm was used as a search strategy and decision trees were used for classification. It was observed that this approach achieved 81.42% accuracy and 6.39% false alarm rate using the UNSW NB-15 dataset.

3. Implementation Strategy

The objective of the proposed approach is to obtain reliable predictions by using an ensemble technique called stacking. The proposed study delineates the results obtained from two datasets captured in two diverse environments:

(i) Binary and multiclass classification results with respect to UNSW NB-15 [28, 29] (an emulated dataset)

(ii) Results obtained using UGR'16 [30] (a cyclostationary dataset formulated through real traffic)

The University of New South Wales Network based 2015 (UNSW NB-15) is a dataset created by a cyber security research group at the Australian Center for Cyber Security [28, 29]. The IXIA Perfect Storm tool was used to capture nine attack categories. This tool incorporates all the updated information needed to include newer attacks from Common Vulnerabilities and Exposures (CVE) site. This dataset has 47 features with two class labels. Tcpdump traces were collected for a span of 31 hours to generate UNSW NB-15 dataset. Since synthetic generation of network traffic was administered to develop this dataset, it failed to trap genuine

behaviors of the Internet [30]. The University of Granada (*UGR'16*) [30] dataset is a more pragmatic attempt made at capturing netflow traces spanning more than four months of network traffic from an Internet service provider (ISP). Founders of this dataset mentioned explicitly that cyclostationary nature of network was considered for the development of this dataset. An important advantage of this dataset is that the background traffic was adequately captured from sensors located in ISP network which normally harbors heterogeneous profiles of clients [30]. This dataset comprising of 16,900 million unidirectional flows offers immense scope to perform extensive experimentation [31].

Figure 1 depicts the stacking framework that comprises base and metaclassifiers, namely, logistic regression (LR), K nearest neighbor (KNN), random forest (RF), and support vector machine (SVM), respectively. The publication of the article Super Learner [32] proclaimed that combination of individual algorithms leads to optimal predictions. Stacking or stacked generalization is a concept proposed by Wolpert [33]. Different machine learning algorithms determine their individual biases on a learning set ultimately filtering out biases. The implementation of a stacked ensemble involves two kinds of models: (i) base models (level 0 classifiers) and (ii) metamodels (level 1 or metaclassifier). The core logic of stacking lies in using the metaclassifier to predict the samples by learning from level 0 classifiers. A significant advantage of the stacking classifier was illustrated, wherein Yan and Han [34] mentioned that stacking can improve the prediction accuracy while considering unbalanced datasets. A study [35] was conducted to emphasize upon the application of artificial intelligence- (AI-) based classifiers. The authors explained that ensembles possess the ability to adapt to the vigorous behaviors of malicious and normal traffic quite effectively. Tables 1 and 2 enumerate the details of network instances considered for experimentation from UNSW NB-15 and UGR'16 datasets, respectively.

3.1. Preprocessing and Selection of Features. Preprocessing was applied to handle miscellaneous data found in the dataset. In order to remove noise and to resolve inconsistencies found in the data, a statistical transformation tool is necessary. In the proposed work, missing values and outliers were compensated by making the distribution normal. However, missing values depend on individual features. While some features may have zero as a missing value, others have zero as part of its value wherever binary data are considered. In order to avoid predicaments, considering relevant features that promise optimal predictions is necessary. Hence, a combination of information gain (IG) and hashing was used to extract the most desirable features. Feature scaling was applied to ensure that those features possessing a greater numeric range do not dominate the ones in smaller numeric ranges. UNSW NB-15 has many features but not all seem to be significant. The essential features were assigned weights in order to prioritise them, and only the best features were extracted. Dimensionality of the features was reduced using hashing technique. It is worthwhile to mention that only eleven features were selected from UNSW NB-15 dataset like *sbytes*, *sttl*, *sload*, *tcprtt*, *smean*, *ct_srv_src*, *ct_state_ttl*, *ct_src_dport_ltm*, *ct_dst_src_ltm*, *ct_srv_dst*, and *service*. Alternatively, the following five features were considered from UGR'16 dataset: *source_ip*, *destination_port*, *forwarding status*, *packets exchanged in the flow*, *and number of bytes*. For a detailed explanation of the abovementioned features and different attack types, [28–30] can be consulted.

3.2. Classification. The critical hyperparameters used for tuning and optimizing the performance of the classifiers are enumerated in Table 3. The strategy to implement the classification framework involved the application of multiple classifiers to resolve the underlying intricacies of data found in both packet-based and flow-based datasets.

Basically, KNN relies on a distance function that computes similarity or difference between two network instances found in the datasets under consideration. The Euclidean distance $d(x, y)$ can be calculated by using the following equation:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}, \quad (1)$$

where x_i refers to the i^{th} feature of the instance x, whereas y_i refers to the i^{th} feature of the instance y. "n" refers to the total number of features found in the dataset. Let $C = \{C_1, C_2, C_3, \ldots C_p\}$. There are "$p$" labels in the dataset. Let "x" be the new sample to be predicted. The objective of KNN classifier is to determine "k" vectors that are close to x. If the majority of the vectors belong to class C_m, then x will be assigned the class label C_m.

Radial basis function (RBF) is a preferred kernel function for many classification problems in machine learning. The following equation defines the RBF:

$$(x, y) = \exp\left(-\frac{\|x - y'\|^2}{2\sigma^2}\right), \quad (2)$$

where $\|x - y'\|^2$ denotes the squared Euclidean distance between two data points x and y. RBF kernel consists of two significant components, namely, gamma and c. Gamma is the decision region. c denotes the penalty for wrongly classifying data points. Whenever "c" is large, SVM will be penalized heavily. The value of c is maintained as 1.0 which indicates that SVM is fairly tolerant of misclassifications that eventually lead to less variance. A higher value when assigned to c can lead to overfitting (Algorithm 1).

4. Results and Discussion

The credibility of any intrusion detection system can be ascertained by four parameters: true positives, true negatives, false positives, and false negatives. True positives denote the correct classifications of normal network instances. True negatives signify the correct classifications of attack samples. False positives indicate the incorrect classification of attack samples into the normal class. False negatives are the normal samples classified as attacks. Some

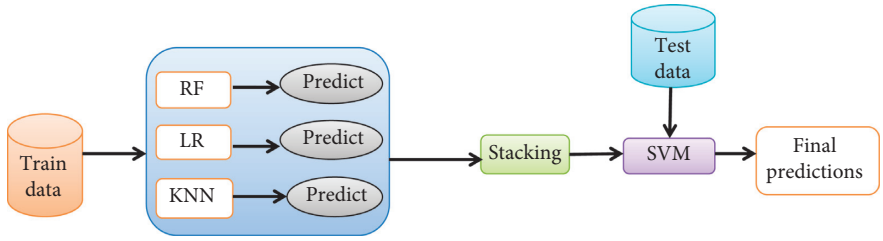

Figure 1: Stacking ensemble.

Table 1: Number of samples considered for experimentation from the UNSW NB-15 dataset.

Type	Training	Testing
Worms	130	44
Shellcode	1133	378
Backdoor	1746	583
Analysis	2000	677
Reconnaissance	10491	3496
DOS	12264	4089
Fuzzers	18184	6062
Exploits	33393	11132
Generic	40000	18871
Normal	56000	37000
Total	**1,75,341**	**82,332**

Table 2: Number of flows considered for experimentation from the UGR'16 dataset.

Type	Count
Blacklist	1,048,576 flows of each attack type
Spam	
SSHscan	
UDPscan	
DOS	
DDOS	
Scan	

Table 3: Critical hyperparameters.

Model	UNSW NB-15	UGR'16
Random forest	Estimators = 100 Criterion = gini	Estimators = 50 Criterion = entropy
K nearest neighbor (KNN)	Neighbors = 5 Metric = Minkowski	Neighbors = 6 Metric = Euclidean
Logistic regression	Penalty = $L2$	Penalty = $L2$
Support vector machine	$C = 1.0$ Kernel = rbf	$C = 1.0$ Kernel = rbf

standard performance metrics defined in the study of network intrusion detection are defined below:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{3}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{4}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{5}$$

$$F1\ \text{score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}, \tag{6}$$

$$\text{false alarm rate} = \frac{\text{FPR} + \text{FNR}}{2}, \tag{7}$$

$$\text{false positive rate} = \frac{\text{FP}}{\text{FP} + \text{TN}}, \tag{8}$$

$$\text{false negative rate} = \frac{\text{FN}}{\text{FN} + \text{TP}}. \tag{9}$$

In order to precisely estimate the efficacy of the proposed approach and corroborate the results obtained from stacked ensemble, both binary and multiclass classification results are presented in this section. Table 4 depicts the results obtained upon classifying the network instances of the UNSW NB-15 dataset into either attack or normal.

In order to testify the predictions and also to affirm that the models do not overfit, mean training accuracy (MTA), mean training precision (MTP), and mean training recall (MTR) values are also mentioned in Table 5.

Table 6 represents actual versus predicted classifications corresponding to each class, namely, normal (N), reconnaissance (R), backdoor (B), denial of service (D), exploits (E), analysis (A), fuzzers (F), worms (W), shellcode (S), and generic (G).

The highest detection rate (recall) of 98.32% is obtained for generic attack type, whereas the least detection rate is reported for backdoor attack type, i.e., 10.79%. However, it is still a challenge to improve the detection rate of attack types like analysis, denial of service (DOS), worms, and backdoor. Precision refers to the relevant results presented by the model.

The netflow traces found in UGR'16 include real background traffic for a substantial duration of four months. The primary reason behind considering this dataset to develop the intrusion detection model can be attributed to the presence of controlled attack traffic that influences the cyclostationary evolution of traffic. Thus the validation of the proposed approach will be more genuine and meaningful using this realistic dataset. 1,048,576 netflow traces of each attack type were considered to comprehend the performance of stacking approach. Figure 2 is a pictorial representation to perceive the performance of stacking ensemble on the UGR'16 dataset by depicting the scores of accuracy, precision, and recall pertaining to different attack types.

Input: Train data $T = \{X_i, Y_i\}_{i=1}^{m}$ $(X_i \in R^n, Y_i \in Y)$
Output: Predictions from the ensemble **E**
Step 1. Impose cross validation in order to prepare a training set for meta-classifier
Step 2. Randomly split T into "m" equal size subsets, i.e., $T = \{T_1, T_2, T_3 \ldots T_m\}$
Step 3. for $m \longleftarrow 1$ to M
  Learn base classifiers namely random forest, KNN, and logistic regression
  for $n \longleftarrow 1$ to N
    Learn a classifier P_{mn} from T or T_m
  End for
Step 4. Formulate a training set for metaclassifier (SVM)
  for each $X_i \in Tm$
  Extract a new instance (x_i', y_i), where $x_i' = \{P_{m1}(X_i), P_{m2}(X_i), P_{m3}(X_i), \ldots, P_{mN}(X_i)\}$
  End for
End for
Step 5. Return $y_i = \{y_1, y_2, y_3, \ldots, y_n\}$ from ensemble

Algorithm 1: Strategy for implementing the stacking ensemble.

Table 4: Binary classification results obtained using the UNSW NB-15 dataset.

TP	TN	FN	FP	Accuracy	Precision	Recall	*F*1 score	AUC	FAR (%)
42535	35365	2797	1635	0.94	0.96	0.93	0.95	0.99	5.2

Table 5: Training results obtained by 10-fold cross validation.

Folds	Training accuracy	Training recall	Training precision
1	0.9291	0.9138	0.9499
2	0.9312	0.9134	0.9516
3	0.9248	0.9092	0.9471
4	0.9286	0.9095	0.9496
5	0.93	0.9112	0.9526
6	0.9327	0.9132	0.9473
7	0.9427	0.9037	0.9475
8	0.9266	0.9094	0.9522
9	0.9285	0.9134	0.9484
10	0.9251	0.9108	0.9510
	MTA: **0.9285**	MTR:**0.9115**	MTP: **0.9497**

Table 6: Multiclass classification results obtained using the UNSW NB-15 dataset.

Index	A	B	D	E	F	G	N	R	S	W	Recall (%)
Analysis	58	0	61	317	32	0	54	1	0	0	11
Backdoor	0	49	79	286	31	1	5	2	1	0	10.79
DOS	3	5	838	2354	66	13	41	11	12	0	25
Exploits	6	6	752	7622	187	33	169	160	25	7	85
Fuzzers	0	2	93	528	2936	7	1217	6	26	0	60.97
Generic	0	2	33	133	14	11512	10	1	2	1	98.32
Normal	19	0	38	163	1260	7	17075	16	16	1	91.82
Reconnaissance	0	2	112	556	5	1	17	2077	2	2	74.8
Shellcode	0	4	6	42	26	0	37	17	184	0	58.22
Worms	0	0	2	18	0	5	0	0	0	15	37.5
Precision (%)	67.44	70	41.6	63.41	64.42	99.42	91.67	90.65	68.65	57.69	

As per the confusion matrix illustrated in Table 7, it is evident that all the seven attack types found in the UGR'16 dataset were differentiated quite aptly by the stacking classifier. The highest attack detection rate was reported for blacklist attack type. It can be noted that this kind of attack detection ability when exhibited by intrusion detection models can prove to be beneficial for counteracting emerging attacks like DDOS, DOS, and scan attacks. Although network instances belonging to the aforesaid attack types are found in conventional datasets like KDD cup 99 and NSL-KDD, such attack traces are definitely obsolete because newer attacks have emerged in recent

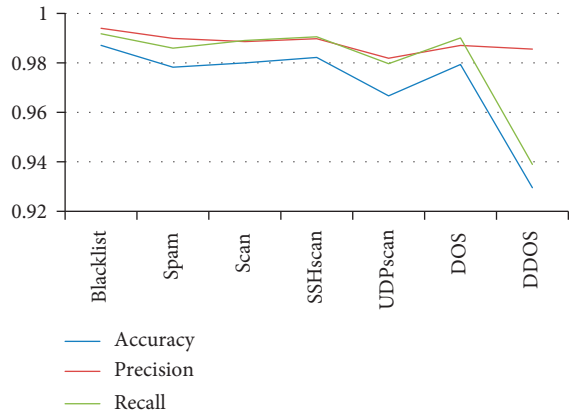

Figure 2: Evaluation metrics of the UGR'16 dataset.

Table 7: Confusion matrix of all the 7 attack types found in UGR'16.

	0	1
Blacklist		
0	944204	5709
1	7834	90829
UDPscan		
0	892295	16466
1	18477	121338
Spam		
0	932187	9532
1	13268	93589
DOS		
0	936949	12311
1	9348	89968
Scan		
0	927854	10710
1	10253	99759
SSHscan		
0	940476	9728
1	8962	89410
DDOS (botnets)		
0	927785	13583
1	60303	46905

years despite similar nomenclature. Table 8 highlights the classwise performance of the seven attack types found in the UGR'16 dataset.

The proposed ensemble model could detect the occurrence of blacklist attack type in the most efficient manner. In order to present reliable results, performance metrics like precision and recall were also considered in addition to accuracy. Recall can be defined as the capability of the intrusion detection model to determine the positive cases correctly, whereas precision refers to the ability of the model to determine the percentage of positive predictions that were correct.

Normally, there is a trade-off that occurs between recall and precision. Since $F1$ score takes into account both precision and recall, it is often used as a performance metric to assess the efficacy of intrusion detection systems. As observable from Table 8, the false alarm rate is considerably low with respect to all the attack categories, and it is an indication that the overall performance of the ensemble model is definitely good. Both false positives and false negatives hamper the performance of network intrusion detection systems. If legitimate traffic is reported as an intrusion, then security analysts may unnecessarily invest their time and resources trying to comprehend a traffic scenario that is absolutely normal. A greater damage is caused when malicious network traffic is identified as normal because such adverse traffic situations may force security experts to overlook some really detrimental traffic scenarios. Any intrusion detection system should not generate too many false alarms. In the current study, the performance of the ensemble model has been considerably good due to the low false alarm rate reported during experimentation. From Table 8, it is obvious that the false alarm rate is quite low pertaining to different attack categories considered in the study.

Typically, receiver operating characteristic (ROC) curve is a pictorial representation of sensitivity vs. 1 – specificity for the entire threshold value. Here, the term sensitivity represents true positives which is projected as a positive rate (which is similar to the recall measurement). It is also written as $P(\text{Pred} = \text{positive} \mid \text{True} = \text{positive})$. Likewise, the term specificity represents $P(\text{Pred} = \text{negative} \mid \text{True} = \text{negative})$. Based on the ratio of true negatives predicted as negatives, ROC curves are used to visualize the relationship between detection rate and false positive rate of a classifier. With respect to the UGR'16 dataset, different attack types have true positive rate around 0.99 and false positive rate ranges between 0.05 and 0.23. Hence, an average value has been obtained for plotting the ROC curve as shown in Figure 3.

Network intrusion detection presents numerous challenges to researchers like recurring cyber attacks, lack of publicly available datasets, and problems associated with benchmark datasets to name a few. Another important parameter for considering an intrusion detection dataset is definitely the kind of network traffic environment used to generate it. Normally, intrusion detection datasets are formulated in either real or emulated network traffic scenarios.

This work has considered two datasets for experimentation (UNSW NB-15 and UGR'16) that are modern in their approach and proposed an ensemble model using supervised machine learning algorithms. Although the nomenclature of attack types found in many intrusion detection datasets is similar, the network traffic environment used to capture the attack traces plays a vital role in deciding whether the intrusion detection framework can be closely modelled to the real world or not. For example, denial of service attack traces are found in KDD cup 99, UNSW NB-15, and UGR'16, but it cannot be generalized that all these attack signatures are similar because they were captured in emulated as well as real network traffic scenarios, respectively, with substantial differences pertaining to attack tools, traffic generators, and test beds [28–31].

Likewise, the credibility of any approach proposed for network intrusion detection can be ascertained by its potential to differentiate between modern attacks (traces of

Table 8: Classwise performance obtained using the UGR'16 dataset.

Metric	Blacklist	Spam	Scan	SSHscan	UDPscan	DOS	DDOS	Overall
Recall	**0.9918**	0.98597	0.98907	0.99056	0.9797128	0.99012	0.93897	0.9809
Precision	**0.9940**	0.98988	0.98859	0.98976	0.9818808	0.98703	0.98557	0.9881
FAR	**0.0054**	0.0062	0.0102	0.0093	0.0147	0.0117	0.0130	0.0101
Accuracy	**0.9871**	0.97826	0.98001	0.98218	0.9666758	0.97934	0.92954	97.19%

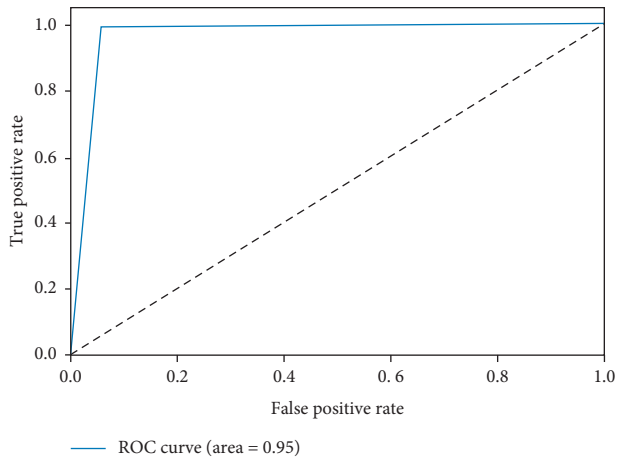

Figure 3: ROC curve obtained for the UGR'16 dataset.

modern attack types are found in UNSW NB-15 and UGR'16). It can be noted that 20 features were used in [27] to achieve the results using the UNSW NB-15 dataset as compared to the proposed approach wherein only 11 features were used to accomplish a superior accuracy and a reasonably lower false alarm rate.

Moreover, as noted in [27], three decision tree classifiers were used to perform classification, but the proposed study employed a diverse set of classifiers to achieve the desired objective quite efficiently. Given the presence of massive network traffic in the real world, it is prudent to consider large number of instances for experimentation as in the case of the UGR'16 dataset. With the advent of Internet of things (IOT), network traffic will only become more and more complex in the coming years [36, 37].

A very negligible false alarm rate has been reported with respect to the UGR'16 dataset, and the least reported false alarm rate is 0.54% pertaining to blacklist attack type. Binary classification results tend to focus only on either normal or attack classification whereby the problem of misclassification between various attack types tends to dissipate, often resulting in a higher accuracy. Hence, multiclass classification becomes indispensable. However, it is still a challenge to improve the attack detection rates of some attack types. Such problems are common while experimenting with multiclass datasets that normally comprise of unbalanced samples. As explained clearly in [38], ensemble of classifiers can be considered as a feasible solution for class imbalance problem. UGR'16 is relatively new and there are no studies pertaining to the implementation of supervised learning algorithms on this dataset. As elaborated in [30], cyclostationary characteristics of the network are well captured by this dataset.

Network traffic, in all possibilities, is cyclostationary because unpredictable fluctuations can be observed that strongly depend on the time of the day and year. As discussed in [30], network traffic exhibits temporal behaviour. In essence, when cyclostationary characteristics are captured by a dataset, it is possible to comprehend the dynamics of network traffic and analyse periodic behaviour. In real world, there is a need to design network intrusion detection systems that take into account cyclostationary features. Apart from the UGR'16 dataset, there is no publicly available dataset at present where cyclostationarity has been captured. Therefore, in order to validate the effectiveness of the proposed approach in a better manner, a dataset that depicts the characteristics of real traffic is also included in the study.

The adoption of network flows in the field of network intrusion detection is extremely important; a missing element in most of the traditional datasets is used for evaluating the performance of intrusion detection systems. An important advantage of the UGR'16 dataset is the presence of unidirectional flows instead of packets unlike the UNSW NB-15 dataset that can be used for decisive anomaly detection. As described in [39], network flows present an aggregated view of the network. Therefore, the time spent to analyse such flows is considerably less. Predominantly, the usage of any flow-based intrusion dataset is advantageous over other datasets because they can be used in a novel manner to detect intrusions in high-speed networks [40].

5. Conclusion

This work has proposed an ensemble approach using the concept of stacking for effective network intrusion detection. Two heterogeneous datasets like UNSW NB-15 (emulated) and UGR'16 (real-time) were used for experimentation. A combination of algorithms, namely, random forest, logistic regression, K nearest neighbor, and support vector machine, resulted in superior predictions with respect to a real-time dataset than an emulated one. The implementation strategy can be further extended to conduct experimentation on different datasets that include recent attack categories. Sophisticated computing engines like Apache Spark can be used in future to increase the processing speed and facilitate scalability for large volumes of network data. From the series of experimentation conducted during the course of this research work, it can be inferred that the proposed approach serves as a competitive perspective for real-time

network intrusion detection. Traffic periodicity and long-term evolution of network traffic cannot be performed using only conventional packet-based intrusion detection datasets. Thus, heterogeneous datasets when applied in the field of network intrusion detection prove to be quite instrumental for gaining better insights into building secure applications.

Data Availability

The datasets used in this work are publicly available for research purposes. (1) https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/ and (2) https://nesg.ugr.es/nesg-ugr16/.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this article.

References

[1] E. Vasilomanolakis, S. Karuppayah, M. Mühlhäuser, and M. Fischer, "Taxonomy and survey of collaborative intrusion detection," *ACM Computing Surveys (CSUR)*, vol. 47, no. 4, pp. 1–33, 2015.

[2] Y. Y. Chung and N. Wahid, "A hybrid network intrusion detection system using simplified swarm optimization (SSO)," *Applied Soft Computing*, vol. 12, no. 9, pp. 3014–3022, 2012.

[3] S. Ganapathy, K. Kulothungan, S. Muthurajkumar, M. Vijayalakshmi, P. Yogesh, and A. Kannan, "Intelligent feature selection and classification techniques for intrusion detection in networks: a survey," *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, no. 1, p. 271, 2013.

[4] A. Aburomman and M. B. I. Reaz, "Review of IDS development methods in machine learning," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, no. 5, pp. 2432–2436, 2016.

[5] E. Bahri, N. Harbi, and H. Nguyen Huu, "Approach based ensemble methods for better and faster intrusion detection," in *Computational Intelligence in Security for Information Systems*, pp. 17–24, Springer, Berlin, Germany, 2011.

[6] P. Yang, Y. Hwa Yang, B. B. Zhou, and A. Y. Zomaya, "A review of ensemble methods in bioinformatics," *Current Bioinformatics*, vol. 5, no. 4, pp. 296–308, 2010.

[7] A. A. Aburomman and M. B. I. Reaz, "A survey of intrusion detection systems based on ensemble and hybrid classifiers," *Computers & Security*, vol. 65, pp. 135–152, 2017.

[8] R. Polikar, "Ensemble learning," in *Ensemble Machine Learning*, pp. 1–34, Springer, Boston, MA, USA, 2012.

[9] R. Sikora, "A modified stacking ensemble machine learning algorithm using genetic algorithms," in *Handbook of Research on Organizational Transformations through Big Data Analytics*, pp. 43–53, IGI Global, Harrisburg, PA, USA, 2015.

[10] G. Zhao, Z. Shen, C. Miao, and R. Gay, "Enhanced extreme learning machine with stacked generalization," in *Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 1191–1198, IEEE, Hong Kong, China, June 2008.

[11] B. Zenko, L. Todorovski, and S. Dzeroski, "A comparison of stacking with meta decision trees to bagging, boosting, and stacking with other methods," in *Proceedings of the 2001 IEEE International Conference on Data Mining*, pp. 669-670, IEEE, Washington, DC, USA, December 2001.

[12] S. Diplaris, G. Tsoumakas, P. A. Mitkas, and I. Vlahavas, "Protein classification with multiple algorithms," in *Panhellenic Conference on Informatics*, pp. 448–456, Springer, Berlin, Germany, 2005.

[13] N. C. Oza and T. Kagan, "Classifier ensembles: select real-world applications," *Information Fusion*, vol. 9, no. 1, pp. 4–20, 2008.

[14] N. Chand, P. Mishra, C. Rama Krishna, E. S. Pilli, and M. Chandra Govil, "A comparative analysis of SVM and its stacking with other classification algorithm for intrusion detection," in *Proceedings of the 2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA) (Spring)*, pp. 1–6, IEEE, Dehradun, India, April 2016.

[15] J. Vanerio and P. Casas, "Ensemble-learning approaches for network security and anomaly detection," in *Proceedings of the Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*, pp. 1–6, ACM, Los Angeles, CA, USA, August 2017.

[16] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," *IEEE Access*, vol. 7, pp. 82512–82521, 2019.

[17] W. Zhang, H. Ren, Q. Jiang, and K. Zhang, "Exploring feature extraction and ELM in malware detection for android devices," in *International Symposium on Neural Networks*, pp. 489–498, Springer, Cham, Switzerland, 2015.

[18] B. Kavitha, D. S. Karthikeyan, and P. Sheeba Maybell, "An ensemble design of intrusion detection system for handling uncertainty using neutrosophic logic classifier," *Knowledge-Based Systems*, vol. 28, pp. 88–96, 2012.

[19] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*, Elsevier, Amsterdam, Netherlands, 2011.

[20] W.-H. Chen, S.-H. Hsu, and H.-P. Shen, "Application of SVM and ANN for intrusion detection," *Computers & Operations Research*, vol. 32, no. 10, pp. 2617–2634, 2005.

[21] M. Govindarajan and R. M. Chandrasekaran, "Intrusion detection using neural based hybrid classification methods," *Computer Networks*, vol. 55, no. 8, pp. 1662–1671, 2011.

[22] A. A. Aburomman and M. B. I. Reaz, "A novel SVM-kNN-PSO ensemble method for intrusion detection system," *Applied Soft Computing*, vol. 38, pp. 360–372, 2016.

[23] R. Rangadurai Karthick, V. P. Hattiwale, and B. Ravindran, "Adaptive network intrusion detection system using a hybrid approach," in *Proceedings of the 2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012)*, pp. 1–7, Bangalore, India, January 2012.

[24] L. Li, Y. Yang, S. Bai, Y. Hou, and X. Chen, "An effective two-step intrusion detection approach based on binary classification and kNN," *IEEE Access*, vol. 6, pp. 12060–12073, 2017.

[25] A. J. Malik, W. Shahzad, and F. A. Khan, "Binary PSO and random forests algorithm for PROBE attacks detection in a network," in *Proceedings of the 2011 IEEE Congress of Evolutionary Computation (CEC)*, pp. 662–668, IEEE, New Orleans, LA, USA, June 2011.

[26] Y.-Y. Zhou and G. Cheng, "An efficient network intrusion detection system based on feature selection and ensemble classifier," 2019, https://arxiv.org/abs/1910.04256.

[27] C. Khammassi and S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," *Computers & Security*, vol. 70, pp. 255–277, 2017.

[28] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set,"

Information Security Journal: A Global Perspective, vol. 25, no. 1–3, pp. 18–31, 2016.

[29] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS)*, pp. 1–6, IEEE, Canberra, Australia, November 2015.

[30] G. Maciá-Fernández, J. Camacho, R. Magán-Carrión, P. García-Teodoro, and R. Therón, "UGR '16: a new dataset for the evaluation of cyclostationarity-based network IDSs," *Computers & Security*, vol. 73, pp. 411–424, 2018.

[31] M. Ring, S. Wunderlich, D. Grudl, D. Landes, and A. Hotho, "Flow-based benchmark data sets for intrusion detection," in *Proceedings of the 16th European Conference on Cyber Warfare and Security*, pp. 361–369, ACPI, Dublin, Ireland, June 2017.

[32] V. D. Laan, J. Mark, E. C. Polley, and A. E. Hubbard, "Super learner," *Statistical applications in genetics and molecular biology*, vol. 6, no. 1, 2007.

[33] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[34] J. Yan and S. Han, "Classifying imbalanced data sets by a novel re-sample and cost-sensitive stacked generalization method," *Mathematical Problems in Engineering*, vol. 2018, Article ID 5036710, 13 pages, 2018.

[35] G. Kumar and K. Kumar, "The use of artificial-intelligence-based ensembles for intrusion detection: a review," *Applied Computational Intelligence and Soft Computing*, vol. 2012, Article ID 850160, 20 pages, 2012.

[36] W. Wang, Y. He, J. Liu, and S. Gombault, "Constructing important features from massive network traffic for lightweight intrusion detection," *IET Information Security*, vol. 9, no. 6, pp. 374–379, 2015.

[37] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in internet of things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, 2017.

[38] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463–484, 2011.

[39] A. Sperotto, *Flow-Based Intrusion Detection*, University of Twente, Enschede, Netherlands, 2010.

[40] M. F. Umer, M. Sher, and Y. Bi, "Flow-based intrusion detection: techniques and challenges," *Computers & Security*, vol. 70, pp. 238–254, 2017.