Data and Metadata. 2024; 3:297
doi: 10.56294/dm2024297

ORIGINAL

A proposed method for detecting network intrusion using an ensemble learning (stacking -voting) approach with unbalanced data

Propuesta de método de detección de intrusiones en la red mediante aprendizaje por conjuntos (stacking -voting) con datos desequilibrados

Anouar Bachar[1] ✉, Omar EL Bannay[1] ✉

[1]Laboratory of Science and Technology for the Engineer, LaSTI-ENSA, Sultan Moulay Slimane University. Khouribga 25000, Morocco.

Cite as: Bachar A, EL Bannay O. A proposed method for detecting network intrusion using an ensemble learning (stacking -voting) approach with unbalanced data. Data and Metadata. 2024; 3:297. https://doi.org/10.56294/dm2024297

Submitted: 20-10-2023 **Revised:** 03-02-2024 **Accepted:** 21-06-2024 **Published:** 22-06-2024

Editor: Adrián Alejandro Vitón Castillo

ABSTRACT

The use of computer networks has become necessary in most human activities. However, these networks are exposed to potential threats affecting the confidentiality, integrity, and availability of data. Nowadays, the security of computer networks is based on tools and software such as antivirus software. Among the techniques used for machine protection, firewalls, data encryption, etc., were mentioned. These techniques constitute the first phase of computer network security. However, they remain limited and do not allow for full network protection. In this paper, a Network Intrusion Detection System (NIDS) was proposed for binary classification. This model was based on ensemble learning techniques, where the base models were carefully selected in a first layer. Several machine learning algorithms were individually studied to choose the best ones based on multiple metrics, including calculation speed. The SMOTE technique was used to balance the data, and cross-validation was employed to mitigate overfitting issues. Regarding the approaches used in this research, a stacking and voting model was employed, trained, and tested on a UNSW-NB15 dataset. The stacking classifier achieved a higher accuracy of 96 %, while the voting approach attained 95,6 %.

Keywords: Binary Classification IDS; Machine Learning; Ensemble Learning; Stacking Model; Voting Model; UNSW-NB15.

RESUMEN

El uso de redes informáticas se ha hecho necesario en la mayoría de las actividades humanas. Sin embargo, estas redes están expuestas a posibles amenazas que afectan a la confidencialidad, integridad y disponibilidad de los datos. Hoy en día, la seguridad de las redes informáticas se basa en herramientas y programas como los antivirus. Entre las técnicas utilizadas para la protección de las máquinas se mencionan los cortafuegos, el cifrado de datos, etc. Estas técnicas constituyen la primera fase de la seguridad de las redes informáticas. Sin embargo, siguen siendo limitadas y no permiten una protección completa de la red. En este trabajo se propuso un sistema de detección de intrusiones en la red (NIDS) para la clasificación binaria. Este modelo se basó en técnicas de aprendizaje ensemble, en las que los modelos base se seleccionaron cuidadosamente en una primera capa. Se estudiaron individualmente varios algoritmos de aprendizaje automático para elegir los mejores en función de múltiples métricas, incluida la velocidad de cálculo. Se utilizó la técnica SMOTE para equilibrar los datos y la validación cruzada para mitigar los problemas de sobreajuste. En cuanto a los enfoques utilizados en esta investigación, se empleó, entrenó y probó un modelo de apilamiento y votación en un conjunto de datos UNSW-NB15. El clasificador por apilamiento alcanzó una precisión superior al 96 %, mientras que el enfoque por votación alcanzó el 95,6 %.

Palabras clave: Clasificación Binaria IDS; Aprendizaje Automático; Aprendizaje Conjunto; Modelo de Apilamiento; Modelo de Votación; UNSW-NB15.

INTRODUCTION

Nowadays, the use of network services has become an indispensable part of most areas of human activity, from personal to professional use. Unfortunately, these networks are exposed to several threats, some of them very costly, especially for businesses.[1] In most cases, these threats target data confidentiality and integrity, as well as the availability of IT network services. To defeat these threats, there are a variety of traditional protection techniques in widespread use, such as anti-virus software, firewalls, data encryption, and more. These techniques represent a first line of security, but they are insufficient to protect fully our network.[2] For this reason, a second line of defense is essential.

Recently, network attacks have become more aggressive due to the architecture of the new tools used to target user workstations and connected terminals. Attackers use highly sophisticated malicious software (malware), making traditional detection tools unable to detect these threats, we're talking about a new generation of attacks, known as "Zero-Day" attacks,[3] previously attacks were targeted at the users of IT systems, e.g. bank customers to steal credit card data, but "Zero-Day" attacks are now capable of threatening large organizations such as hospitals, banks, etc. According to the "2023 Global Threat Intelligence Report"[4] published by "NTT Security Holdings", we find that these attacks menace very critical sectors, either by organized criminals or by individual attackers. According to this report, the five most threatened sectors are:

- The information technology (IT) sector (25 %)
- Transport and distribution sector (8,12 %)
- The manufacturing sector (19,01 %)
- Education sector (11,37 %)
- Public sectors (9,10 %)
- Other sectors (26,50 %)

According to the above, setting up an effective intrusion detection system for companies is a crucial necessity, and machine learning is considered among the most robust techniques for designing a high-performance IDS.

In this work, we have proposed a binary classification model based on an ensemble learning approach (stacking-voting), using three classifiers carefully selected from a first layer. In this layer, we trained and evaluated nine different classifiers, the XGB Classifier (XGBC), Decision Tree Classifier (DTC), Extra Tree Classifier (ETC), Gradient Boosting Classifier (GBC), KNeighbors Classifier (KNC), AdaBoost Classifier (ADC), SGDC Classifier (SGDC), Random Forest Classifier (RFC), and Logistic Regression (LRC), XGBC, ETC, and SGDC were selected as the base models, and the LRC model was selected as the meta-estimator for our stacked model.

Previous Work of Machine Learning applications in intrusion detection system field

Natesan et al.[5] proposed a NIDS using the KDDCUP 99 dataset. They used a hybrid model combining two classification models, naive bayesian, and decision tree, achieving a detection rate of 85,78 %.

Depren et al.[6] proposed a NIDS, they achieved 99 % TPR with KDD-CUP99, also Zhou et al. [7] used the same data source and proposed an efficient model based on the distance method, it obtained a very high detection rate (TPR=99,54 %), Divyatmika et al.[8] used the MLP (Multilayer Perceptron) algorithm with KDD-CUP 99, the detection rate achieved was 99 %, finnaly Bachar et al.[9] used SVM with two kernels polynomial and guassian, a detection rate of 94 % was achieved with the UNSW-NB15 dataset.

Some works are based on an ensemble learning approach for the detection of intrusions in networks, this approach can improve either the speed of detection or the rate of correct predictions.

We begin with Shen et al.[10] used an ensemble learning method for NIDS, three public datasets are used for their proposed model: NSL, Kyoto, and KDDCUP 99.

An adaptive ensemble learning method was used by Gao et al.[11] for NIDS, the proposed model was evaluated with NSLKDD Test+, and they achieved a detection rate of 85,2 %.

Hsu et al.[12] proposed an ANIDS model based on AE_SVM (Autoencoder with a single SVM class) and RF, this model is evaluated with both datasets UNSW-NB15 and NSLKDD, and they achieved a correct prediction rate of 91,8 % for UNSW-NB15 and 91,7 % for NSLKDD.

For an optimal classification result, UÇAR et al.[13] they tested several algorithms as base predictors and the SVM model is used as meta-estimator for the different stacked models, the 4 stacked models are tested with NSLKDDTest+ and NSLKDDTest 21 data, the best prediction rate is scored for the model based on DT,ANN, LR as base predictors and SVM as meta-estimator, they achieved 90 % in the best case.

In the paper by Das et al.[14] the ensemble learning is used for the evaluation of a NIDS with several data sources, they used the basic predictors LR, DT, NB, NN, and SVM, concerning the performances, they achieved 85,7 % as detection rate.

The base estimators GNB (Guasian Naive Bayes), LR and DT are used with the meta-estimator SGD (Stochastic Gradient Descent) by Thockchom et al.[15] the proposed model reached 93,88 % with UNSNW-NB15.

Tama et al.[16] proposed an anomaly-based IDS. This model has two levels of classification aggregated by a

https://doi.org/10.56294/dm2024297

majority vote, these two levels are based on Rotation Forest and bagging, and they achieved a detection rate of 91,27 % with the UNSW-NB15 dataset.

In the paper by Immanuel et al.[17] an ensemble learning method is used for NIDS, using the two estimators RF and KNN as base models, and to obtain a final prediction, they use RF as a meta-model, this approach is tested with the NDSL-KDD dataset, obtaining a correct prediction rate of 89,98 %.

Using machine-learning technics, several datasets are used to detect computer network intrusions. We start with the DRAPPA98 dataset collected by MIT University in 1998.[18] Then the KDD99 dataset was collected in 1999.[19] This data source contains a variety of records concerning four attack classes (DOS, Probe, R2L, U2R). However, it does have some issues, notably the presence of a very large number of duplicate records, which affects IDS performance.[20] To improve intrusion classification results, an improved version of KDD99 was built to have the NSLKDD.[21] The last two datasets are widely used in machine learning for NIDS. On the other hand, other datasets are less widely used, such as CAIDA, CSIC, CISDA, and DARPA 2009.

In recent years, for network intrusion classification, the NSLKDD and KDDCUP 99 datasets have been widely used. However, several works such as[20,22,23] have shown that these datasets have major problems such as the lack of sufficient data representation for detecting recent attacks (zero-day). These data sets contain redundant data that does not provide sufficient coverage of real and recent attacks, which is why in this paper we have selected another, more powerful data source called UNSW-NB15.[24]

METHOD

Main Contribution

This work is articulated around the search for an optimized classification model in terms of speed and detection rate, in this paper we propose an evaluation of nine different classifiers tested on the UNSW-NB15 dataset, and a comparative study has been conducted based on several metrics (accuracy, precision, Recall, and F1-score). In the first level, we selected the three best estimators with high performance, then in the second level we proposed an ensemble learning model based on the three classifiers selected in the first level, then we compared the three ensemble learning approaches (Stacking, Soft voting and Hard voting) proposed with other studies, a K-folds cross-validation was used to avoid overfitting problems[25] the SMOTE technique (Synthetic Minority Over-sampling Technique) is applied to solve data imbalance problems.

Intrusion Detection System (IDS)

An IDS is an essential tool for detecting threats on a network. It analyzes incoming and outgoing packets to detect various intrusions. We classify the IDS according to its location in the network and its detection method. Concerning the disposition of the IDS, we talk about NIDS (Network Intrusion Detection System) placed before or after a firewall.

This NIDS secures the entire network. On the other hand, the HIDS (Hote Intrusion Detection System) provides security at the machine level only. The second classification concerns the method of detection[26] the first method is IDS signature-based, this type of IDS detects intrusions based on a known signature. It is effective in detecting classic attacks. However, this IDS cannot detect new attacks. On the other hand, it generates a reduced number of false alarms especially if it is placed after a firewall figure 1. The major problem with this detection method is the inability to discover recent attacks (0-day attacks) having an unknown signature.[27,28] The second detection method is the anomaly-based IDS. This IDS builds a profile of normal behavior to compare with any malicious activity on the networks.[29] However, this type detects recent attacks but generates a high FAR (False Alarm Rate) compared to the signature-based IDS.

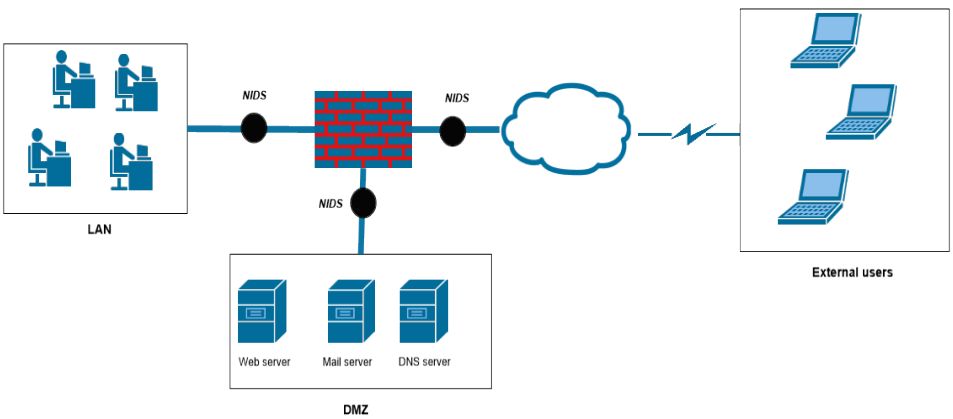

Figure 1. IDS Placement

Proposed architecture

In this work, we proposed a NIDS model figure 2, the UNSW-NB15 dataset was used to experiment with our model, a data preprocessing phase (Replacement of missing values, encoding of categorical values, data normalization) is needed, also the SMOTE technique was applied to balance the data, in this study we evaluated nine supervised individual models, Based on their performances, we have selected the best three estimators as inputs for the next level, we then combined the selected models using a stacking, soft-voting, and hard-voting technique, to return a highly accurate final prediction.

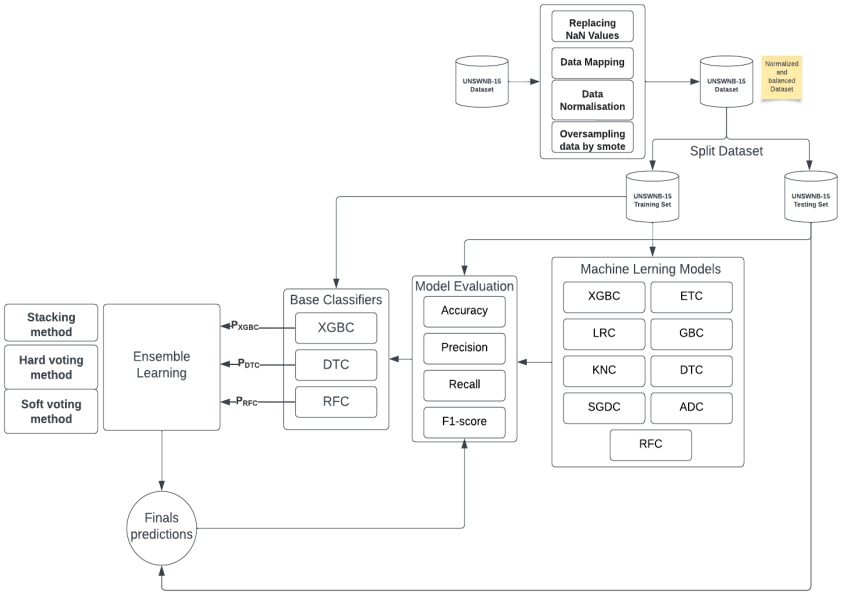

Figure 2. Proposed model

UNSW-NB15 dataset

It is a public dataset developed by "the Australian Centre for Cyber Security" (ACCS) using "IXIA Perfect Storm" tools. The UNSW-NB15 dataset contains 49 features representing nine categories of attacks table 1 and one feature for the target label (attack or non-attack).[30] The UNSW-NB15 records are structured into 6 data types of features: Flow, Basic, Time, Additional features, and Label target.

Table 1. Category of attacks on UNSW-NB15

Type	Description
DOS	A type of attack that targets a service's availability, it consists of sending packets in masse into a network to block access to a service requested by a legitimate client.
Fuzzers	Using an automated tool, the cybercriminal sends random test data to an application or operating system to identify potential vulnerabilities.
Backdoors	A passive attack based on a malware installed on a victim's computer, this program exploits system vulnerabilities to grant unauthorized access to the attacker for further attacks.
Analysis	Generally, this type of attack precedes other attacks, involving the collecting of information about the target.
Generic	An attacker targets a machine without any previous knowledge, for example decoding an encrypted text by trying all possible combinations.
Reconnaissance	collecting information about a target, such as hardware type, installed system, and version, etc.
Shellcode	A binary code represented by a string, this code is injected into memory to exploit system vulnerabilities to allow a hacker to control the infected machine.
Exploit	When an attacker exploits a known vulnerability in an operating system or software.
Worms	This is infectious software capable of spreading through the network between different connected machines.

Data Preprocessing

Before training our models, a pre-processing phase is necessary, consisting of replacing missing values, normalizing numerical data, and encoding categorical features. Since the target in our dataset (attack class -

and non-attack class) is unbalanced, the SMOTE method will be used to solve this issue.

Data-Mapping: we cannot use textual data in Machine Learning models, so a transformation of nominal data into numerical data is a required step. For example, the service feature may contain alphabetical data such as (FTP, SNMP, HTTP, DNS...), so we transform these values into numerical values (1,2,3,4 ...).

Normalizing numerical data: data normalization is putting numerical data in the 0 to 1 range to increase processing speed. We use the Min-Max method as shown in equation 1.

$$X_{normalised} = \frac{X - min(X)}{max(X) - min(X)} \quad (1)$$

Data balancing with SMOTE

The target class in the UNSW-NB15 dataset (attack; no-attack) is unbalanced, this issue is normal given the nature of the traffic collected by IXIA PERFECT STORM, in reality, the benign incoming traffic is higher than the malicious traffic figure 3, this imbalance leads to a misclassification of the minority classes (malicious traffic), resulting in inaccurate predictions. The SMOTE technique[31] can solve this problem by adding synthetic records to our dataset figure 3 while maintaining the consistency of the dataset.

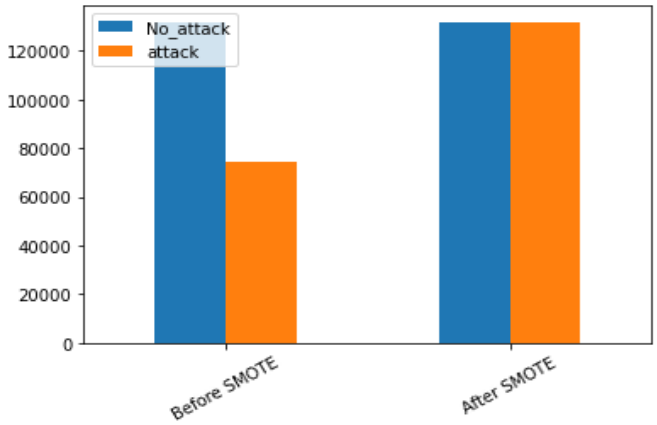

Figure 3. Classes Distribution before and after SMOTE

Machine Learning algorithms

Machine learning algorithms provide an important possibility for the design of strong NIDS[32] while supervised algorithms achieve good results for the detection of known intrusions, but do not cover unknown attacks.[33] In our study, we will train several single classifiers, these are XGBC, DTC, LRC, ETC, GBC, KNC, ADC, SGDC, and RFC. These models are evaluated to use the best of them in a second ensemble Learning model. Our approach concerns a binary classification problem capable of detecting whether an attack occurs.

Ensemble Learning methods

Ensemble learning is a powerful method capable of combining multiple classifiers, producing a model with high prediction and low bias. In this paper, we use a Voting method and a Stacking method.

Stacking method

Among the ensemble learning models, we find the stacked model[34] which involves the creation of a powerfully trained metamodel based on the predictions of several classifiers. In our method, we used the three basic estimators XGBC, DTC, and RFC as independent trained models in the first phase, then the metamodel combines the predictions of these three models to give a final prediction (figure 4).

Voting method

This method is similar to the stacking method, the only difference being that the voting-based method doesn't require a meta-estimator combining the predictions of the underlying classes, so the final prediction is the aggregation of majority voting based on the predictions of the classifiers at level 1 figure 4 , In our study, we implemented two voting approaches (hard voting and soft voting), the difference being that the hard voting is based on strict majority voting. In contrast, soft voting is the average probability for each predicted class.

https://doi.org/10.56294/dm2024297

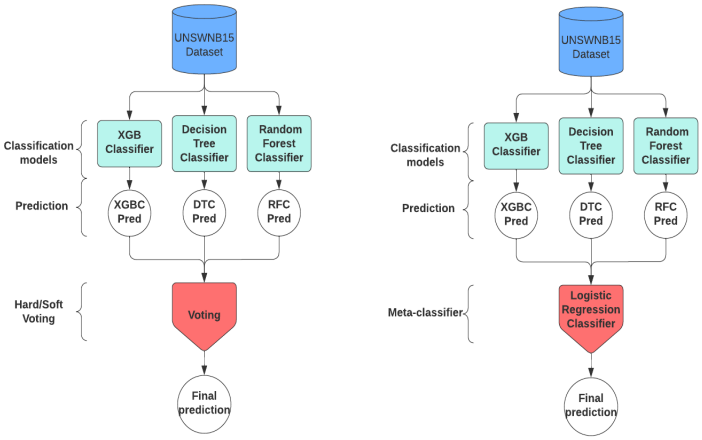

Figure 4. Stacking and voting method

Cross-validation

To improve our proposed architecture, we used a cross-validation technique. This is a very practical technique for reducing overfitting problems[35] since our model will be trained on several subsets of the data, and the cross-validation also improves the accuracy of the new data.

Evaluation metrics

Confusion matrix

To evaluate our proposed NIDS, we focus on two important concepts (reliability and relevance), the reliability of an IDS means that a good IDS must generate an alert for each actual threat, on the other hand, the relevance implies that an IDS should not generate an alert for normal incoming traffic.

All the evaluation metrics used in this study come from the confusion matrix, (CM) which is a two-dimensional matrix containing information on the actual class and the predicted class[36] table 2 explains the different cases for an IDS.

Table 2. Confusion matrix

	Predicted Attack	**Predicted No-Attack**
Actual Attack	TP (True Positive) (Actual Attack - Alarm produced)	FN (False Negative) (Actual Attack – No alarm)
Actual No-Attack	FP (False Positive) (No attack – alarm produced)	TN (True Negative) (No attack – No alarm produced)

To evaluate our model, we used 4 evaluation metrics: Precision, **Accuracy,** Recall, and F1-score. Table 3 describes these metrics:

Table 3. The evaluation metrics

Metric	**Formula**	**Description**
Accuracy	$\frac{TP+TN}{TestSet\ size}$	Measures the capability of a NIDS to recognize the true predictions for both classes (attack and no-attack). A good IDS must have a very high Accuracy.
Precision	$\frac{TP}{TP+FP}$	Defined as a ratio of TP to the aggregate value of both TP and FP rate.
Recall	$\frac{TP}{TP+FN}$	Measure how often our model correctly detects the positive instances (TP) from all the actual positive samples in our dataset. The recall is the proportion of true positive (TP) cases which are rightly classified
F1-score	$\frac{2*TP}{2*TP+FP+FN}$	The harmonic mean of precision and recall, which is very important to establishing an agreement between the precision and the recall.

RESULTS

In this section, we present the results that we have achieved, all models are trained and evaluated with the publicly available UNSW-NB15 dataset, the first subsection concerns the evaluation metrics used to experiment with our NIDS, In the second subsection we will present the detailed results of the binary classification of the nine single models and the performances of the stacking, soft voting, and hard voting models, the last subsection involves the comparison of our proposed models with exiting study that's using an ensemble learning technic for network intrusion detection.

Machine learning models evaluation

This sub-section is reserved for the results we have obtained, our models are evaluated according to the evaluation metrics previously mentioned in table 3, table 4 shows the results obtained for the nine classifiers, and figure 5 illustrate the fitting time for each classifier.

Table 4. Machine learning models performances

Classifier	Precision	Accuracy	Recall	F1-score	Fitting time (per seconde)
XGBC	0,95	0,958	0,95	0,95	2,47
DTC	0,94	0,935	0,94	0,95	7,67
LRC	0,90	0,895	0,90	0,90	2,98
ETC	0,95	0,949	0,95	0,95	44,71
GBC	0,93	0,932	0,93	0,93	265,4
KNC	0,92	0,912	0,91	0,91	22,26
ADC	0,93	0,922	0,92	0,92	51,85
SGDC	0,89	0,893	0,89	0,89	2,73
RFC	0,95	0,951	0,95	0,95	100,13

Table 4 clearly shows that the XGBC model achieves the best accuracy (95,8 %), precision (95 %) and execution speed (2,47s), while the classifiers (RFC, ETC and DTC) perform well in terms of accuracy and precision. The only weakness of RFC is its long execution time (100,13s).

The GBC classifier achieves 93,2 % accuracy and 93 % precision, but this model is considered as the longest-fitting classifier (265,4s), while the other classifiers (ADC, KNC and LRC) achieve a medium accuracy.

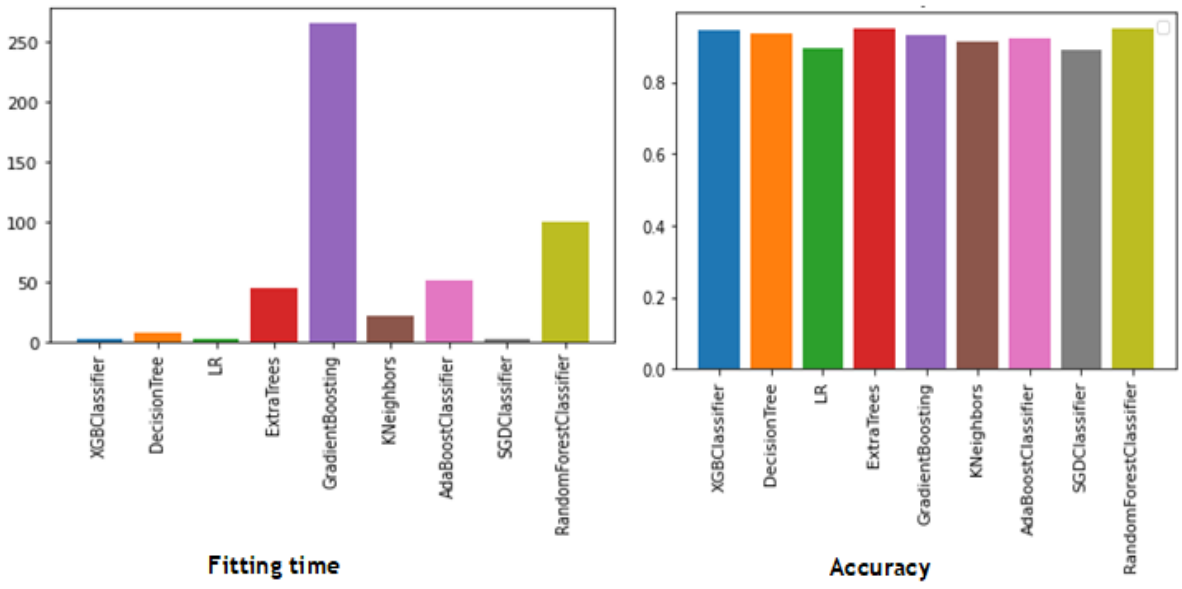

Figure 5. Fitting time and accuracy machine learning models comparison

https://doi.org/10.56294/dm2024297

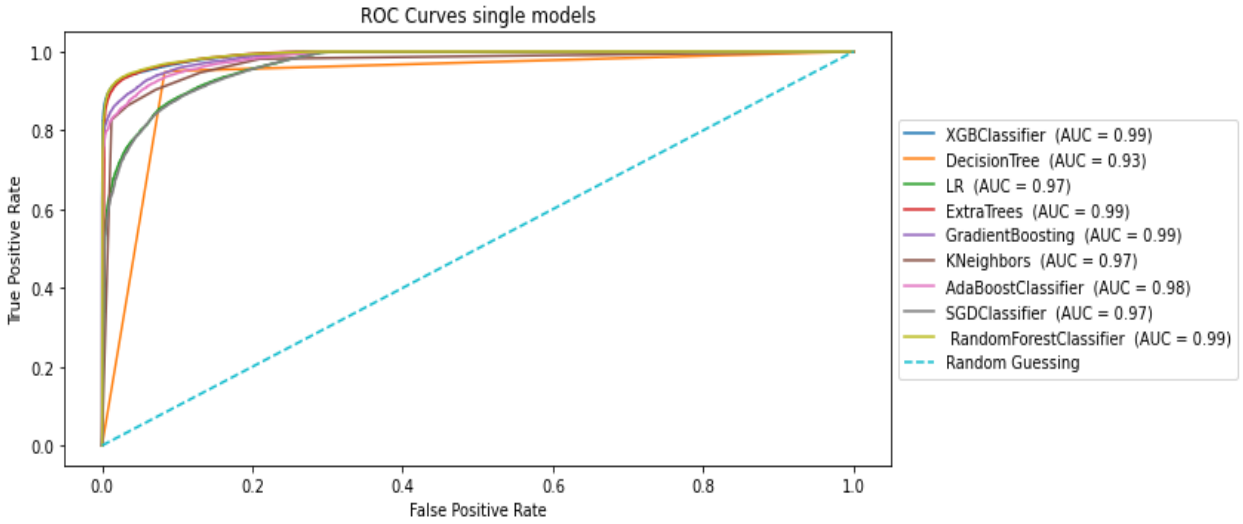

Figure 6. ROC Curves for single ML models

Ensemble learning models evaluation

For the second level of training and to achieve optimum performance, we used the best-performing classifiers (XGBC, RFC, and DTC), which are combined for powerful prediction. The three methods (Stacking, Hard-voting, and soft-voting) are tested on the UNSWNB-15 dataset using a cross-validation cv=10 figure 7, the table 5 summarizes all the results obtained for the different methods.

Table 5. The evaluation metrics of a classification model

Model	Accuracy	Precision	Recall	F1-score
ENSEMBLE USING Stacking	0,960	0,95	0,95	0,95
ENSEMBLE USING Soft-voting	0,954	0,95	0,95	0,95
ENSEMBLE USING Hard-voting	0,956	0,95	0,95	0,95

We observed that ensemble learning methods improved the detection rate by reducing the number of false negatives and false positives (see confusion matrix in figure 7). In this study, the best model was the Stacking model, which raised the relevance of our IDS to 96 %.

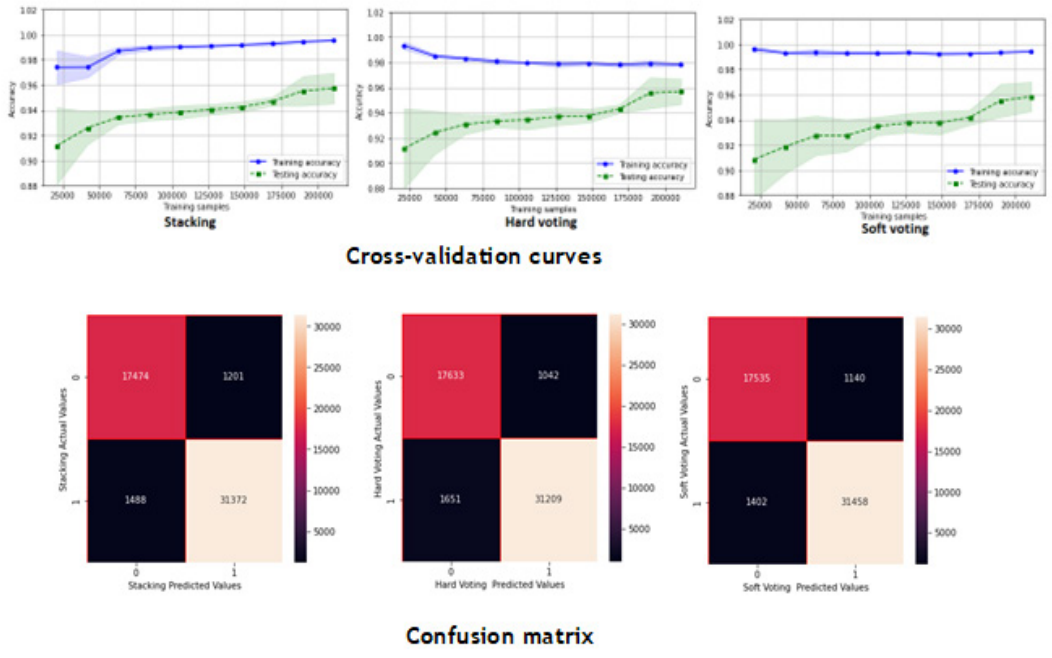

Figure 7. Cross-validation curves and Confusion Matrix for Stacking, Hard Voting, and Soft Voting models

https://doi.org/10.56294/dm2024297

Performance comparative analysis

In this subsection, we compare our obtained results with the different works based on the ensemble learning approach. Table 6 summarizes the various results of the different studies related to our research.

Table 6. Comparison with state-of-the-art study

Model	Base estimators	Meta-Estimators	Accuracy
Ensemble Voting[11]	DT,RF, KNN, DNN et MultiTree	-	85,2 %
Ensemble Voting[14]	LR,DT, NB, NN, et SVM		85,7 %
Ensemble Stacking[15]	GNB, LR et DT	SGD	93,88 %
Ensemble Stacking[12]	AE_SVM et RF	Weight combnaison	91,8 %
Hard voting[16]	Stage 1: Rotation Forest Stage 2: Bagging	-	91,27 %
Ensemble Stacking[17]	RF, KNN	RF	89,98 %.
Stacking (our prposed model)	XGBC, ETC et SGDC	LRC	96 %
Soft voting (our prposed model)	XGBC, ETC et SGDC	-	95,4 %
Hard voting (our prposed model)	XGBC, ETC et SGDC	-	95,6 %

The comparative evaluation of existing models with our proposals shows the effectiveness of ensemble methods for classification. Existing Voting models have shown acceptable performance, but the Stacking model[15] has surpassed the others with an accuracy of 93,88 %. Our proposed models performed better than all the others, with an accuracy of 96 % for the Stacking model and around 95-96 % for the Voting models. We attribute this superior performance to the right combination of machine learning algorithms and the use of appropriate meta-estimators. In conclusion, this study confirms the crucial importance of ensemble methods and offers valuable directions for the development of high-performance NIDS.

CONCLUSIONS

Computer networks are a target for attackers, and NIDS is a very practical mechanism for protecting our network. In this paper, we have proposed a model based on two methods (stacking and voting), these methods are based on base classifiers selected from an investigation of several single models, we have evaluated these classifiers according to precision, accuracy, recall and F1-score, also we have taken into consideration the complexity and execution time of each model. XGBC, DTC, and RFC were selected for their effectiveness (accuracy = 95 %) as base classifiers for Hard-Voting, Soft-Voting, and Stacking models. Otherwise, LRC was chosen as the stacking model meta-estimator due to its fast execution time (2,98 s). Employing the SMOTE technique ensured data balance, and all models were trained on the UNSW-NB15 dataset. In summary, this study confirms the crucial importance of ensemble methods and offers valuable guidance for the development of high-performance NIDS.

BIBLIOGRAPHIC REFERENCES

1. Maglaras LA, Kim KH, Janicke H, Ferrag MA, Rallis S, Fragkou P, et al. Cyber security of critical infrastructures. Vol. 4, ICT Express. Korean Institute of Communication Sciences; 2018. p. 42–5.

2. Choo KKR. The cyber threat landscape: Challenges and future research directions. Comput Secur. 2011 Nov;30(8):719–31.

3. Guo Y. A review of Machine Learning-based zero-day attack detection: Challenges and future directions. Vol. 198, Computer Communications. Elsevier B.V.; 2023. p. 175–85.

4. NTT DATA. Global Threat Intelligence Report. 2023.

5. Natesan P. Multi Stage Filter Using Enhanced Adaboost for Network Intrusion Detection. International Journal of Network Security & Its Applications. 2012 May 31;4(3):121–35.

6. Depren O, Topallar M, Anarim E, Ciliz MK. An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. Expert Syst Appl. 2005 Nov;29(4):713–22.

7. (Communications in Computer and Information Science 259) Ju-Sung Kang, Dowon Hong (auth.), Tai-hoon Kim, Hojjat Adeli, Wai-chi Fang, Javier García Villalba, Kirk P. Arnett, Muhammad Khurram Khan (eds.

8. Divyatmika, Manasa S. A Two-tier Network based Intrusion Detection System Architecture using Machine Learning Approach. In: International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT). 2016. p. 42–7.

9. Bachar A, El Makhfi N, EL Bannay O. Machine learning for network intrusion detection based on SVM binary classification model. Advances in Science, Technology and Engineering Systems. 2020;5(4).

10. Shen Y, Zheng K, Wu C, Zhang M, Niu X, Yang Y. An Ensemble Method based on Selection Using Bat Algorithm for Intrusion Detection. Computer Journal. 2018 Apr 1;61(4):526–38.

11. Gao X, Shan C, Hu C, Niu Z, Liu Z. An Adaptive Ensemble Machine Learning Model for Intrusion Detection. IEEE Access. 2019;7:82512–21.

12. Hsu YF, He ZY, Tarutani Y, Matsuoka M. Toward an online network intrusion detection system based on ensemble learning. In: IEEE International Conference on Cloud Computing, CLOUD. IEEE Computer Society; 2019. p. 174–8.

13. UÇAR M, UÇAR E, İNCETAŞ MO. A Stacking Ensemble Learning Approach for Intrusion Detection System. Düzce Üniversitesi Bilim ve Teknoloji Dergisi. 2021 Jul 31;9(4):1329–41.

14. Das S, Saha S, Priyoti AT, Roy EK, Sheldon FT, Haque A, et al. Network Intrusion Detection and Comparative Analysis Using Ensemble Machine Learning and Feature Selection. IEEE Transactions on Network and Service Management. 2022 Dec 1;19(4):4821–33.

15. Thockchom N, Singh MM, Nandi U. A novel ensemble learning-based model for network intrusion detection. Complex and Intelligent Systems. 2023 Oct 1;9(5):5693–714.

16. Tama BA, Comuzzi M, Rhee KH. TSE-IDS: A Two-Stage Classifier Ensemble for Intelligent Anomaly-Based Intrusion Detection System. IEEE Access. 2019;7:94497–507.

17. V.J. Immanuel Jeo Sherin, Dr.N. Radhika. Stacked Ensemble-IDS Using NSL-KDD Dataset. J Pharm Negat Results. 2022 Jan 1;13(SO3).

18. DARPA98 Dataset. Available on: https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset. DARPA 98.

19. KDDCUP99. Available on: https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. 1999. KDD99.

20. Tavallaee M, Bagheri E, Lu W, Ghorbani AA. A detailed analysis of the KDD CUP 99 data set. In: Proc IEEE Symp Comput Intell Secur Defense Appl. 2009. p. 1–6.

21. NSL-KDD(2009) NSL-KDD | Datasets | Research | Canadian Insti tute for Cybersecurity | UNB, https://www.unb.ca/cic/datasets/nsl.html.

22. Moustafa N, Slay J. 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. 2016 Apr 4;25(1–3):18–31.

23. Moustafa N, Slay J. The Significant Features of the UNSW-NB15 and the KDD99 Data Sets for Network Intrusion Detection Systems. In Institute of Electrical and Electronics Engineers (IEEE); 2017. p. 25–31.

24. UNSW-NB15. Available on: https://research.unsw.edu.au/projects/unsw-nb15-dataset [Internet]. 2015. UNSW-NB15.

25. Kohavi R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection [Internet]. Available from: http//roboticsStanfordedu/"ronnyk

26. Bijone M. A Survey on Secure Network: Intrusion Detection & Prevention Approaches. American Journal of Information Systems [Internet]. 2016;4(3):69–88. Available from: http://pubs.sciepub.com/ajis/4/3/2

https://doi.org/10.56294/dm2024297

27. Kabiri P, Ghorbani AA. Research on Intrusion Detection and Response: A Survey [Internet]. Vol. 1, International Journal of Network Security. 2005. Available from: http://isrc.nchu.edu.tw/ijns/

28. Buczak AL, Guven E. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications Surveys and Tutorials. 2016 Apr 1;18(2):1153–76.

29. Mischiatti M, Neri F. Applying Local Search and Genetic Evolution in Concept Learning Systems to Detect Intrusion in Computer Networks.

30. Moustafa N, Slay J. UNSW-NB15: A Comprehensive Data set for Network Intrusion Detection systems (UNSW-NB15 Network Data Set) [Internet]. Available from: https://cve.mitre.org/

31. Chawla N V, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: Synthetic Minority Over-sampling Technique. Vol. 16, Journal of Artificial Intelligence Research. 2002.

32. Prasad R, Rohokale V. Springer Series in Wireless Technology [Internet]. Available from: http://www.springer.com/series/14020

33. Laskov P, Düssel P, Schäfer C, Rieck K. Learning intrusion detection: Supervised or unsupervised? In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2005. p. 50–7.

34. Ting KM, Witten IH. Issues in Stacked Generalization. Vol. 10, Journal of Artiicial Intelligence Research. 1999.

35. Cawley GC, Talbot NLC. On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. Vol. 11, Journal of Machine Learning Research. 2010.

36. Deng X, Liu Q, Deng Y, Mahadevan S. An improved method to construct basic probability assignment based on the confusion matrix for classification problem. Inf Sci (N Y). 2016 May 1;340–341:250–61.

FINANCING

The authors did not receive financing for the development of this research.

CONFLICT OF INTEREST

None.

AUTHORSHIP CONTRIBUTION

Conceptualization: Anouar Bachar, Omar EL Bannay.
Research: Anouar Bachar, Omar EL Bannay.
Drafting - original draft: Anouar Bachar, Omar EL Bannay.
Writing - proofreading and editing: Anouar Bachar, Omar EL Bannay.