Ensemble based Effective Intrusion Detection System for Cloud Environment over UNSW-NB15 Dataset

Uzma Amin[1], Aamir S Ahanger[2], Faheem Masoodi[3], Alwi M Bamhdi[4]

Department of Computer Science, University of Kashmir, India[1,2,3]

Department of Computer Science, Umm AL-Qura University, Saudi Arabia[4]

Corresponding author: Faheem Masoodi, Email: masoodifahim@uok.edu.in

Advanced computing innovations are rapidly evolving, resulting in the advent of new organizational and operational strategies. Cloud computing has emerged as one of the pre-eminent innovation in the recent years. Cloud computing enables its clients to access flexible, distributed computing domain via internet. Cloud has manifested itself as a viable framework that facilitates the use of application domains, data and infrastructural facilities that mainly encompasses workstations, network and storage infrastructure. Regardless of robust and comprehensive server processing capabilities in contrast to client's processing capabilities and efficiency there are numerous security risks to the cloud from both outside and within the cloud that might exploit security flaws to cause damage. Traditional security measures have some flaws when it comes to completely shielding the networks and devices from increasingly advanced attacks. Consequently, it is all important to build an intrusion detection system to detect and prevent all kinds of intrusions in the cloud with high accuracy along with low false alarms. In this study we have suggested an anomaly-based intrusion detection system that employs ML algorithms for detection of unknown malicious attacks using an ensemble approach over the UNSW-NB15 dataset. The experimental output demonstrated the accuracy of 99.28% and 99.47% for random forest and bagging algorithms respectively.

Keywords: Cloud Environment, Intrusion, UNSW-NB15 Dataset.

2021. In Raju Pal & Praveen K. Shukla (eds.), *SCRS Conference Proceedings on Intelligent Systems*, 483–494. Computing & Intelligent Systems, SCRS, India.
https://doi.org/10.52458/978-93-91842-08-6-46

1 Introduction

A cloud environment is a collection of resources for leveraging multiple consumers with on-demand services. The Cloud computing enables consumers to access numerous services along with data storage as well as computational resources with minimal data overhead. The National Institute of Standards and Technology (NIST) described cloud computing as a framework for delivering ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (such as virtual machines, services, applications, networks, & storage) that could be instantly provisioned and released with very less managerial efforts or service provider interaction [1].

Cloud storage enables its clients to save data to a remote database in place of keeping it on an exclusive hard drive or rather to a local storage device. Because of its simplicity people are relocating one's data along with application software to cloud data centres. The downside of cloud computing is that by outsourcing, clients get deprived of physical control over their files and they transfer the control to an untrustworthy cloud service provider or third party [2]. Regardless of robust and comprehensive server processing capabilities in contrast to client's processing capabilities and efficiency, there are numerous security risks to the cloud from both outside and within the cloud that might exploit security flaws to cause damage. These vulnerabilities can compromise data confidentiality, data integrity, and data availability. The figure 1 depicts the most common security vulnerabilities and issues connected with data storage.

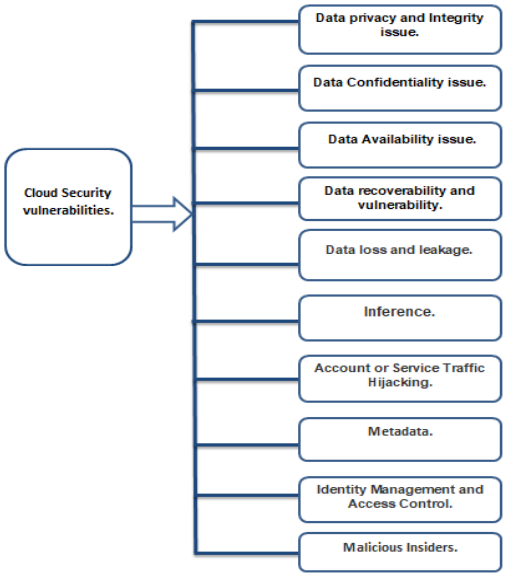

Fig. 1. The most common security vulnerabilities and issues connected with data storage

Traditional security measures example firewalls, encryption, user authentication and access control in data transmission have some flaws when it comes to completely shielding networks and devices from increasingly advanced attacks. Talking about firewalls; even if the Firewall is properly configured, there are still numerous security issues. For instance; if a firewall is placed to permit packets from specific networks, an intruder from outside the firewall can enter false source address to avoid this check. 75% of all intrusions originate outside the firewall; however, threats originating within the firewall, such as from disgruntled employees, are typically the most destructive. Outsider intrusions can be detected by firewalls, but insider intrusions cannot be identified [3].

Furthermore, firewalls are incapable of dealing with a whole new class of attacks. The basic idea behind a firewall is to keep intruders out and prevent confidential data from getting out. Unfortunately, there are intruders who attempt to flood the target with legitimate packets, causing it to collapse; such an attack is referred to as DoS [4]. The sent packets usually have false source addresses, making it difficult to track down the intruder. Another type of DoS attack is in which the intruder jeopardises hundreds of computers around the world and then instructs them all to attack the same target at the same time. This sort of intrusion is known as a DDoS (Distributed Denial of Service) attack. There are still many attacks that the firewall is unable to handle.

When deciding which packets to allow through the firewall, the firewall examines the IP, UDP, TCP and ICMP header fields. Contrarily Deep Packet Inspection is required to determine several types of intrusions, particularly those that the firewall cannot detect. It is therefore essential to develop IDS that is adequate enough to detect as well as prevent both inside and outside attacks in the cloud environment. An intrusion detection system (IDS) is a software or hardware system that monitors and analyses activities in a device or a network to detect signs of data breaches [5]. The goal of intrusion detection systems (IDS) is to ensure the security of a network or system with regard to confidentiality, integrity, and availability. A firewall is typically the first line of defence in a network, and IDS is used when there are signs of an intrusion that the firewall was unable to block. As the number and severity of attacks has increased, intrusion detection systems (IDSs) have become an essential component of most organisations' security infrastructure. Intrusion detection methods are divided into three categories [6].

- Rule-Based Intrusion Detection.
- Signature-based IDS.
- Anomaly-based IDS (AIDS).

Rule-Based Intrusion Detection: In this approach, predefined rules are kept in the IDS knowledge base. When an event occurs, it is processed according to the rules that have been saved. If the event satisfies the rule, it is classified as a regular occurrence; otherwise, it is classified as an intrusion. The rules in rule-based intrusion detection must be updated on a regular basis, which is a time-consuming process and can be considered as a drawback [7].

Signature-based IDS: In this particular approach signatures of malicious activities are maintained in an intrusion detection knowledge base in signature-based intrusion detection. Each intrusion has its own set of malicious signatures, such as unsuccessful logins, failed file and folder access, unsuccessful application attempts and data packet type. These signatures are used by signature-based intrusion detection systems to detect and prevent future intrusions [8]. If the signatures match, it is considered an intrusion; otherwise, it is considered a normal occurrence. Signature-based intrusion detection's effectiveness is determined by the signatures stored in the database. As a result, in order to improve performance, an increasing number of signatures should be stored in the IDS knowledge base, which is considered a disadvantage of this detection technique.

Anomaly-Based Intrusion Detection: Anomaly detection is a method for detecting abnormal patterns that do not follow the normal behaviour. Deviation from the normal is examined and analysed in this method. If the deviation from normal behaviour is significant, the occurrence is referred to as intrusion. Their ability to detect novel attacks makes them appealing. Another benefit is that normal activity profiles are customized for each device, programme, or network, making it difficult for attackers to figure out the behaviours they may engage in.

On the basis of the deployment, there are primarily four different kinds of IDS employed in Cloud scenario [9]: Host based Intrusion Detection System (HIDS), Network based Intrusion Detection System (NIDS), Hypervisor based Intrusion Detection System, and Distributed Intrusion Detection System**.** HIDS keeps track of and analyses data from a single host machine. HIDS collects information about processes and files associated with a given host's software environment to identify intrusion on the machine. HIDS monitors changes in the file system, kernel, and database

activity on the host computer. Any change in usual behaviour indicates the existence of an attack whereas an intrusion detection system (IDS) that is network-based detects intrusions by tracking traffic across network devices [10]**.** NIDS analyses network traffic for malicious behaviour including DoS attacks, port scans, and data breaches. For intrusion detection, the data obtained from the network is analysed against the known intrusions. The Hypervisor-based IDS allows analysing the available data in order to identify unethical activities by examining the communication among the VM and the Hypervisor, as well as multiple VMs within the virtual network and finally DIDS is integration of several intrusion detection systems (IDS), such as HIDS and NIDS. They interact over a large network with many other IDS or a central server, allowing for better network management.

The most frequent obstacles encountered by existing techniques are that several IDSs still have a high false alarm rate, creating numerous warnings for low-risk scenarios, increasing the burden on security analysts and potentially causing severe attacks to go unnoticed. Another issue with traditional IDSs is that they are incapable of detecting novel intrusions. Since network environment changes rapidly, new attack variants and novel intrusions appear on a regular basis [11]. Consequently, it is all important to build IDS to detect and prevent such intrusions in the cloud with high detection and low false alarms. Thereby, effective intrusion detection in cloud involves the implementation of modern intelligent techniques like Machine Learning (ML) techniques. The capability of machine learning algorithms to deal with true as well as unknown data makes them ideal for detecting intrusions. These approaches improve the precision and effectiveness of anomaly detection and signature-based IDS. In this paper we have suggested an ensemble method aimed at increasing the accuracy of intrusion detection in cloud computing.

2 Literature Review

Gao et al. [12] suggested a new ensemble methodology (FSSL-EL) for detection of intrusions in cloud environment. The proposed framework combined ensemble approach along with fuzziness-based technique. The suggested model in the first stage creates an ensemble model based on CART as base learner for the labelled data. With the object of combining the output of CARTs, a 3- layer neural network was applied by the authors. Second for the unsupervised part, a fuzziness-based approach was incorporated to explore the inner structure of unlabelled data and finally the same ensemble technique was used to merge supervised and unsupervised approaches. The outcome exhibited that the proposed approach acquired an accuracy of 71.29% and 84.54% on KDD test 21 and KDD test+ datasets respectively.

With the intention of creating a more capable learner, the authors in [13] proposed a voting ensemble model of 3 base classifiers: decision tree, K-nearest neighbour and logistic regression. For their dataset they have used NSL-KDD dataset. The detection rate that they obtained was higher in case of user to root attack which was twice as high as that of existing approaches.

Garg et al. [14] presented a hybrid technique for network anomaly detection that uses GWO and CNN. For effective network anomaly detection, the suggested model contains two stages: In the first stage improved GWO is employed for selection of essential features. In the second stage, improved CNN is used to classify network anomalies. The obtained outcomes demonstrated that the suggested cloud-based anomaly detection approach outperforms existing models.

A methodology was suggested by Kasongo et al. [15] in which they identified and analysed various malicious activities using the UNSW-NB15 dataset. So as to reduce the complexity, dimensionality reduction was applied with the help of XG Boost algorithm; thereby 19 ideal attributes were chosen. Subsequently following algorithms were employed for detection of intrusions: Support Vector Machine, k-Nearest-Neighbour, Logistic Regression, Artificial Neural Network and Decision Tree. The efficacy of algorithms was evaluated by comparing the result acquired when all the 49 features of UNSW-NB15 dataset were taken into consideration to that when only 19 ideal features were

selected and moreover performance of algorithms was also compared to the pre-existing models. The analysis demonstrated that by applying XG Boost for dimensionality reduction the decision tree enhanced its accuracy from 88.13% to 90.85%.

Jing et al. [16] presented a non- linear scaling methodology using SVM so as to detect intrusions. UNSW-NB15 dataset has been used in this study with the aim of training and testing the model. The binary and multiclass classifications were taken into account in this study. The outcome obtained for binary classification exhibited the accuracy of 85.99% and false positive ratio of 16.50%. Furthermore, for multiclass classification the accuracy of 75.77% was obtained and false positive ratio of 3.04%. The authors further compared the efficacy of SVM with many other classifiers and it was observed that the proposed model was more efficacious for detection of intrusions.

In [17] Aljamal et al. Presented a hybrid approach for detection of unethical activities in cloud environment. For this particular study UNSW-NB15 dataset has been employed. The authors implemented dimensionality reduction with the help of principle component analysis. The proposed approach contains two modules. In first module clustering model was created whose output serves as input to the second module i.e., SVM. The acquired result demonstrated an accuracy of 88.6% for the clustering module and 84.7% for the SVM.

3 Proposed Approach

In our paper, we have proposed an anomaly-based intrusion detection system that employs ML algorithms for detection of unknown malicious attacks using an ensemble approach that integrates the benefits of each single detection algorithms.

Advantages of using ensemble model are:

- The likelihood of overfitting is minimized as the scenario where one classifier is dominative is reduced.
- Models which split the job into multiple smaller segments are much more likely to encapsulate patterns that a single model might overlook.

Dataset

UNSW-NB15 dataset has been used in this study with the aim of training and testing the model. The dataset was released in 2015 [18]. A total of 2,540,044 records are present. In comparison to previous benchmark datasets such as KDDCUP 99, this dataset has a more complex structure [19]. As a result, this particular dataset is more comprehensive in terms of assessing the existing network intrusion detection systems. Each record in the dataset has 49 features categorized into five data types: integer, float, nominal, timestamp, and binary. The obtained features were divided into six groups: basic features, content features, flow features, time features, labelled features and additional generated features.

Dataset Attack Categories: The problem of categorising occurrences into more than two categories is known as multi-class classification. The Attack-category has been created in UNSW-NB15 dataset to classify the records into particular classes. As shown in table 1, the UNSW-NB15 dataset encompasses nine attack categories:

Dataset Packets distribution: There are 2 540,044 records in total. A part of this dataset is split into two sets: Training and testing. In the test dataset and the train dataset, there are 175,341 and 82,332 records, respectively. The fig.2. depicts the distribution of legitimate and malicious records in the train and test set.

Table 1. Attack categories in UNSW-NB15 dataset

Type	No. of records	Description
Normal	2,218,761	Natural transaction data.
Fuzzers	24,246	Attempting to cause a program or network suspended by feeding it the randomly generated data.
Analysis	2,677	It contains different attacks of port scan, spam and HTML files penetrations.
Backdoors	2,329	A technique in which a system security mechanism is bypassed stealthily to access a computer or its data.
DoS	16,353	A malicious attempt to make a server or a network resource unavailable to users, usually by temporarily interrupting or suspending the services of a host connected to the internet.
Exploits	44,525	The attacker knows of a security problem within an operating system or a piece of software and leverages that knowledge by exploiting the vulnerability.
Generic	215,481	A technique works against all block-ciphers (with a given block and key size), without consideration about the structure of the block-cipher.
Reconnissance	13,987	Contains all strikes that can simulate attacks that gather information.
Shellcode	1,511	A small piece of code used as the payload in the exploitation of a software vulnerability.
Worms	174	A computer worm is a type of malware that spreads copies of itself from computer to computer. A worm can replicate itself without any human interaction, and it does not need to attach itself to a software program in order to cause damage.

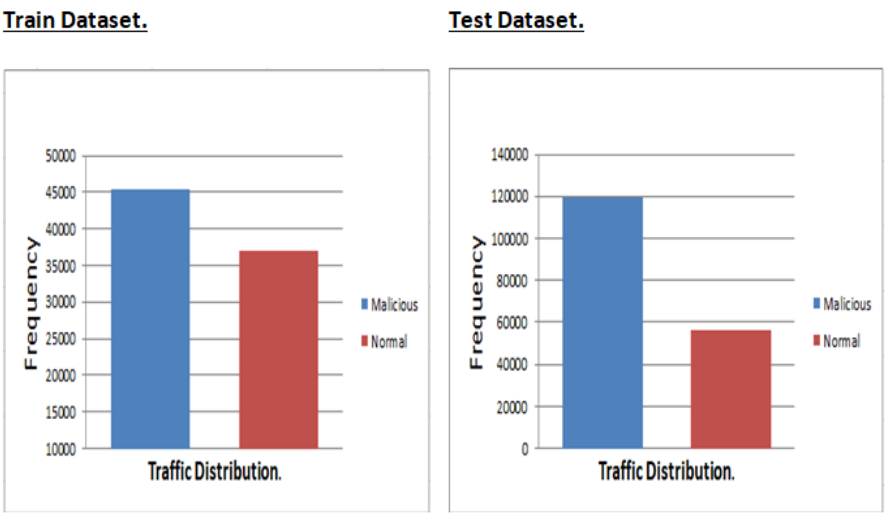

Fig. 2. Traffic Distribution in UNSW-NB15 dataset

Dataset Pre-Processing: Prior to pre-processing, the dataset features were analysed using various EDA techniques aiming to achieve a deep insight of:

- The properties of the data: statistical features, schema and so on.
- The quality of the data: missing values, data types that are inconsistent, and so on.
- The predictive power of the data: for example, the correlation between attributes and the target.

In this paper, the training dataset, which has 82,332 records, and the testing dataset, which has 175,341 records, were combined (concatenated). Afterwards, a 75–25 split was adopted, with 75 percent of the records chosen at random used for training and 25 percent for testing. This extra step was taken to provide more training information to the model.

Data pre-processing is one of the most essential aspect in the data-driven method (machine learning). Generally, the data acquired is not adequate or suitable for use in a machine learning task. As a result, data pre-processing is desirable to obtain better results. When we have the following traits in our data, we must consider data pre-processing before performing a machine learning task, such as:

Missing values: Missing values arise when a feature in a record has no data value recorded for it.

Redundant Records: When a set of records in a dataset are duplicated, it is referred to as redundant records.

Nominal features: Sometimes referred to as categorical features. It often refers to any categorical feature that reflects discrete values that belong to a finite set of categories or classes (i.e., no numerical data).

Non similar scale features: That is, we must ensure that all features have the same value ranges. For example, if the value range of the attribute A is [0-10000] while the range of the attribute B is [0-10]. In this scenario, we consider normalising the data to ensure that all of the feature's scale in the same way.

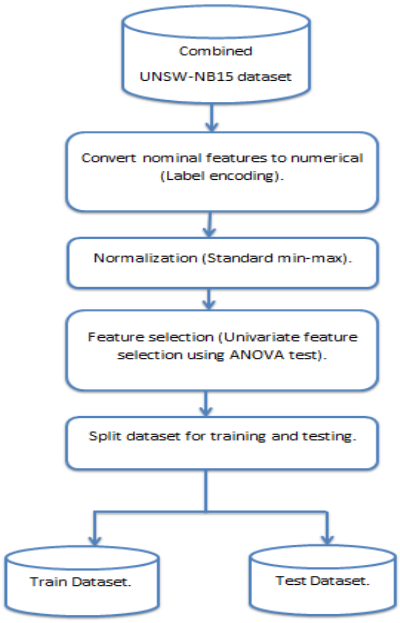

Fig. 3. Data preparation flowchart

In the data pre-processing step, there are a number of other factors to consider. In UNSW-NB15 dataset neither missing values nor redundant records exist. However, all 49 features are not

required to be used. There are four attributes in the flow features (source and destination IP addresses, source and destination port numbers) that we can't utilise in a machine learning technique as they'll bias the model toward those addresses; therefore it's better to eliminate them [20].

There are a few nominal attributes in the UNSW-NB15 dataset. Three of the 49 non-target features of UNSW-NB15 are nominal (protocol, service and state). Because most machine learning classifiers as well asscalers can only deal with numerical quantity, we convert the nominal attributes to numerical. Moreover, thefeatures of this particular dataset do not have the same value range. We applied label encoding and feature scaling (also known as data normalisation) to overcome these two problems.

However, the concern is to decide which one to use to normalise our data. The ideal approach is to choose it based on empirical findings (i.e., the technique that produces the better results is the most appropriate for the nature of our data). We applied the Min-Max method to normalise the UNSW-NB15 dataset features because it preserves the relationships between the original data values.

Feature Selection:

Univariate feature selection was applied for feature selection by using Anova test.

Univariate feature selection chooses the ideal features using univariate statistical tests. Every attribute is compared to the target feature in order to see whether there is a significant relation among them. When we examine the correlation among one attribute and the target feature, we overlook the other attributes, for this reason it is referred to as 'univariate. Every single attribute has a test score associated with it. Eventually, the test score of each and every attribute is compared, and the attributes with the highest scores are chosen.

Out of the 46 features generated by combining the training and testing datasets, univariate feature selection determined the score for every individual attribute indicating the correlation among the attribute and the target feature. Based on the obtained scores we selected the important 39 features.

Machine learning Classifier:

Building the ensemble model: With the aim of enhancing the accuracy of the Intrusion detection system, we trained individual classifiers as base learners and developed an ensemble classifier based on them and a voting technique was used to make classification decisions.

The machine learning classifiers: Bagging algorithm and Random Forest were used to assess the efficacy of anomaly-based IDS over the UNSW-NB15 dataset in this paper.

Bagging algorithm:

Bagging, or bootstrap aggregation, is a simple but effective ensemble strategy for addressing unstable classification issues. When employing the bagging approach in an ensemble, all of the ensemble's algorithms are used in parallel. Bagging is based on the principle of using bootstrapped samples of the original dataset to train various classifiers (base learners/models). Decision trees are used as base classifiers. Subsets will be created by randomly selecting records from the training dataset and replacing them [21]. On each of these bootstrapped subsets, we would then train distinct classifiers. Each of these base classifiers will identify the problem's class label. So, after training is done, we will provide the test data to the model. Each classifier would be given a test data set and they will provide us the output accordingly. This is where we integrate all of the base models' predictions [22]. This stage is known as the aggregation stage and finally we will use a voting classifier. Voting classifier will consider the class that receives the greatest number of votes as output.

As all algorithms run in parallel, each one can be run on a separate processor to speed up the process. This is a significant advantage as compared to the boosting method as multicore CPUs are

now widely used in personal computers. When compared to a single algorithm, the ensemble architecture does not considerably increase processing time because the only extra time required is used for the decision function that integrates the outputs of all algorithms.

Random Forest (RF):

RFs are ensemble classifiers that are applied to classify as well as analyse intrusion detection data [23]. The dissimilarity that separates bagging from random forest is that in random forests, just a subset of attributes is chosen randomly from the whole collection of features and the best split attribute from the subset is applied to split each node in a tree, whereas in bagging, all attributes are taken into account while a node is being split [24].

RF is a collection of base estimators, usually decision trees that are amalgamated to frame a random forest. Within a forest every single decision tree is created using a bootstrap sample from the original data and a subgroup of randomly chosen attributes (input variables) based on the best split in the sub node creation [25]. Bootstrapping is a statistical resampling approach in which a dataset is randomly sampled with replacement. Since the sub-samples and features are selected at random, the correlation between the estimators is minimised, resulting in a more accurate model. In addition, combining many base estimators results in a robust model with lower generalisation error than a single estimator.

Each tree is trained with a randomly selected subset of records from the original training data and output class labels that have obtained the maximum votes [26]. RF achieves exemplary classification accuracy and also deals with data outliers, noise and moreover it is less prone to overfitting. The RF has higher accuracy than Adaboost and it is less likely to over fit.

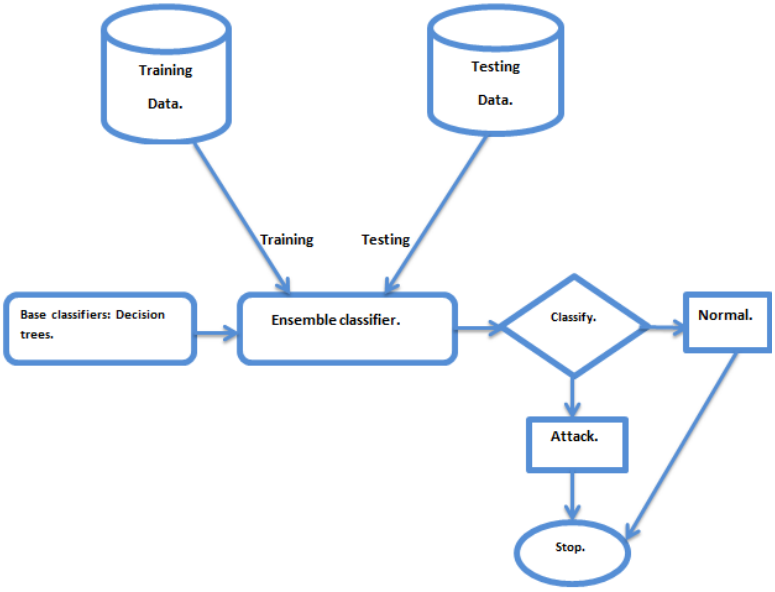

Fig. 4. Ensemble model framework

4 Results

Training and testing of classifiers were done on 257,673 instances of UNSW-NB15 dataset and 39 ideal features were taken into consideration. We have used the standard metrics like accuracy,

confusion matrix, recall, F1-Score to assess the outcomes of the classifiers-Random Forest and bagging algorithm for binary classification. In intrusion detection the positive class denotes malicious activity, while the negative class denotes normal traffic. The results obtained are shown in the table 2:

Table 2. Confusion matrix for random forest and Bagging algorithm

- Confusion matrix for Random Forest:

	Positive	Negative
Positive	22802	82
Negative	378	41157

- Confusion matrix for Bagging algorithm:

	Positive	Negative
Positive	22962	119
Negative	218	41120

Experimental Results:

Table 3. Experimental outcomes of ensemble algorithms with binary classification

Classifier Name	Accuracy	Precision	F1-Score	Recall
Random Forest (Label 0)	99.28%	98%	99%	100%
Random Forest (Label 1)	99.28%	100%	99%	99%
Bagging Algorithm (Label 0)	99.47%	99%	99%	99%
Bagging Algorithm (Label 1)	99.47%	100%	100%	99%

Graphical representation of results is depicted in fig. 5.

5 Conclusion

This paper suggested an IDS that incorporates feature selection and ensemble approach. For training and testing purpose UNSW-NB15 dataset was used with binary classification. For feature selection we have employed univariate feature selection using ANOVA test and out of 44 independent features generated by combining the training and testing datasets univariate feature selection selected the important 38 features for model building.

The ensemble classifiers employed for intrusion detection are: bagging and random forest algorithms that made use of decision tree as the base classifiers. The experimental outcomes demonstrated that the models have high accuracy than the existing models that made use of UNSW-NB15 dataset for detection of intrusions. The accuracy achieved are 99.28% and 99.47% for random forest and bagging algorithms respectively.

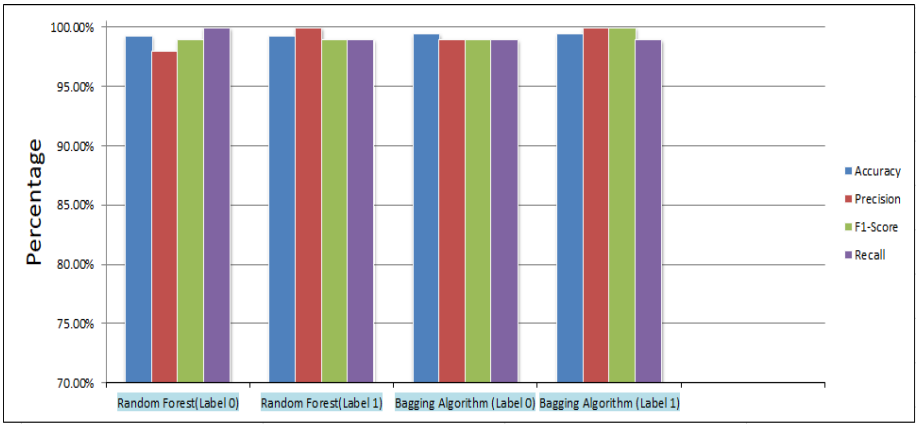

Fig. 5. Graphical representation of results

References

[1] Mell, P. et al. (2011). The NIST-National Institute of Standards and Technology- Definition of Cloud Computing. *NIST Special Publication*, 7:800-145.

[2] Bamhdi, A. M., Abrar, I. and Masoodi, F. (2021). An ensemble based approach for effective intrusion detection using majority voting. *Telkomnika,* 19(2):64–671.

[3] Shamshirband, S. et al. (2020). Computational intelligence intrusion detection techniques in mobile cloud computing environments: Review, taxonomy, and open research issues. *Journal of Information Security and Applications*, 55:102582.

[4] Bokhari, M. U. and Masoodi, F. (2012). BOKHARI: A new software oriented stream cipher: A proposal. In the *WICT,* 128–131.

[5] Masoodi, F., Alam, S. and Bokhari, M. U. (2011). SOBER Family of Stream Ciphers: A Review. *International Journal of Computer Applications*, 23(1):1–5.

[6] Bhati, B. S. and Rai, C. S. (2020). Analysis of Support Vector Machine-based Intrusion Detection Techniques. *Arabian Journal for Science and Engineering,* 45:2371–2383.

[7] Abrar, I., Ayub, Z. and Masoodi, F. (2021). Current Trends and Future Scope for the Internet of Things. In book: *Internet of Things in Business Transformation: Developing an Engineering and Business Strategy for Industry 5.0,* 185–209.

[8] Masoodi, S. T. et al. (2019). Security and privacy threats, attacks and countermeasures in Internet of things. *International Journal of Network Security & Its Applications,* 11(2):67–77.

[9] Modi, C. et al. (2013). A survey of intrusion detection techniques in Cloud. *Journal of Network and Computer Applications*, 36(1):42-57.

[10] Ahanger, A. S., Khan, S. M. and Masoodi, F. (2021). An Effective Intrusion Detection System using Supervised Machine Learning Techniques. In the *Proceeding of 5th ICCMC,* 1639–1644.

[11] Teli, A. T. and Masoodi. (2021). Security Concerns and Privacy Preservation in Blockchain based IoT Systems: Opportunities and Challenges. In the *ICICNIS*, 29–36.

[12] Gao, Y. et al. (2018). A Novel Semi-Supervised Learning Approach for Network Intrusion Detection on Cloud-Based Robotic System. *IEEE Access*, 6:50927-50938.

[13] Masud, M. R. A. and Mustafa, H. (2019). Network Intrusion Detection System UsingVoting Ensemble Machine Learning. In the *Proceedings of the 3rd IEEE International Conference on Telecommunications and Photonics.*

[14] Garg, S. et al. (2019). A Hybrid Deep Learning-Based Model for Anomaly Detection in Cloud Datacenter Networks. *IEEE Transactions on Network and Service Management*, 16(3):924-935.

[15] Kasongo, S. M. and Sun, Y. (2020). Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset. *Journal of Big Data,* 7:105.

[16] Jing, D. and Chen, H. (2019). SVM Based Network Intrusion Detection for the UNSW-NB15 Dataset. In the *13th International Conference on ASIC*, 1-4.

[17] Aljamal, I. et al. (2019). Hybrid Intrusion Detection System Using Machine Learning Techniques in Cloud Computing Environments. In the *IEEE 17th International Conference on Software Engineering Research, Management and Applications*, 84-89.

[18] Moustafa, N. and Slay, J. (2015). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In the *Military Communications and Information Systems Conference*, 1-6.

[19] Janarthanan, T. and Zargari, S. (2017). Feature selection in UNSW-NB15 and KDDCUP'99 datasets. In the *IEEE 26th International Symposium on Industrial Electronics*, 1881-1886.

[20] Pandow, B. A., Bamhdi, A. M. and Masoodi, F. (2020). Internet of Things: Financial Perspective and Associated Security Concerns. *International Journal of Computer Theory and Engineering*, 12(5):123–127.

[21] Masoodi, F. et al. (2021). Machine Learning for Classification analysis of Intrusion Detection on NSL-KDD Dataset. *Turkish Journal of Computer and Mathematics Education,* 12(10):2286–2293.

[22] Masoodi, F. S. and Bokhari, M. U. (2019). Symmetric Algorithms I. *Emerging Security Algorithms and Techniques*, 79–95.

[23] Pham, N. T. et al. (2018). Improving performance of intrusion detection system using ensemble methods and feature selection. In the *Proceedings of the Australasian Computer Science Week Multiconference*, 2.

[24] Ahmad, I. et al. (2018). Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection. *IEEE Access*, 6:33789-33795.

[25] Teli, T. A. and Masoodi, F. (2021). Blockchain in Healthcare: Challenges and Opportunities. *SSRN Electronics Journal*, 1–6.

[26] Abrar, I. et al. (2020). A Machine Learning Approach for Intrusion Detection System on NSL-KDD Dataset. In the *Proceeding of ICOSEC,* 919–924.