Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 1 : 2024
ISSN : **1906-9685**

Journal of Nonlinear Analysis and Optimization : Theory & Applications
Editors-in-Chief :

A MACHINE LEARNING-BASED INTRUSION DETECTION SYSTEM

[#1]**TALLURI UPENDER,** ***Research Scholar,***
[#2]**Dr. BECHOO LAL,** ***Professor & Guide,***
[#1,2]***Department of Computer Science & Engineering,***
[#1,2]**SHRI JAGDISHPRASAD JHABARMAL TIBREWALA UNIVERSITY, RAJASTHAN.**
[#3]**Dr. REGONDA NAGARAJU,** ***Professor& Co-Guide,***
[#3]***Department of CSE-AI&ML, School of Engineering,***
[#3]**MALLA REDDY UNIVERSITY, HYDERABAD, TELANGANA.**

ABSTRACT: The internet has brought about enormous and far-reaching change in the world. Indeed, it helps people keep their social relationships and contact with others in those networks when they need help. Indeed, both firms and individuals face numerous risks when exchanging personal and professional information. Given the internet's critical significance in our daily lives, the security of our data is continually jeopardized. IDS is critical for protecting internet users from potentially harmful network attacks. An Intrusion Detection System (IDS) is a network monitoring system that detects and identifies suspicious or unauthorized activity. It sends notifications whenever an event occurs. This research will focus on three areas: machine learning and the algorithms NB, SVM, and KNN. The first phase will involve applying these strategies to determine the best level of precision using the USNW NB 15 DATASET. The second stage of our database processing uses the most efficient algorithm, which is determined by the outcomes of the previous stage. To assess the model's performance, we will run tests on two different datasets. To determine the efficacy of the suggested approach, performance is examined using the NSL-KDD and UNSW-NB15 datasets.

Index terms: *Machine Learning, Algorithms NB, SVM AND KNN.*

1. INTRODUCTION

The spread of computing devices has been rapid. Many people rely on laptops, desktop computers, tablets, and cellphones as essential tools in their everyday lives. The primary focus here is on the safety of data gathered via the internet; Intrusion Detection Systems (IDS) are used to assure the security of data transported across networks.

An intrusion detection system (IDS) is a hardware or software component that analyzes network or system activity for malicious activities or policy breaches. It then generates reports for the management system. There is no doubt that an intrusion detection system is required, so it is critical to create an accurate model. Machine learning has proven to be an effective investigative tool in this domain, capable of detecting every unexpected event that occurs in any system's traffic. A successful intrusion detection system (IDS) must be highly proficient in detecting malicious network traffic. The effectiveness of categorization algorithms is heavily influenced by their accuracy.

This paper describes a unique intrusion detection system (IDS) approach that improves the accuracy and efficiency of detecting hostile network activity. The first section presents our dataset, which will be trained by three different classification algorithms; the second section presents our dataset, which will be trained by the higher accuracy tree algorithms listed below; and the final section contains a conclusion and some issues that have been brought to our attention for future research.

2. DATASET DESCRIPTIONS

Scientists can acquire a wide range of publicly available datasets from the internet. An examination of the literature revealed that numerous papers were written decades ago and would not be particularly useful in identifying novel threats. Some of the datasets accessible include KDD98 and KDD'99.

Slay N. M. (2016) reports that the UNSW-NB15 dataset was developed in 2015 at the Australian Centre for Cybersecurity's cyber range lab. The dataset is available in a variety of forms, including CSV. We opted not to use the original CSV files due to the high amount of records (about 2.5 million) and the split of these entries into four different files.

We conduct our research using the training and testing sets from the revised CSV files, which contain 82,332 items and 175,341 transactions. The dataset has 47 features, including nominal, category, and numerical data types. The dataset is binary and has multiple labeled classes. Table 1 shows the distribution of each assault across the training and testing sets.

Table 1. The UNSW-NB15 Datasets collection.

DATASET	CLASS	TRAIN-SET	TEST-SET
UNSW-NB15	NORMAL	56 000	37 000
	GENERIC	40 000	18 871
	EXPLOITS	33 393	11 132
	FUZZERS	18 184	6 062
	DOS	12 264	4 089
	RECONNAISS ANCE	10 491	3 496
	ANALYSIS	2 000	677
	BACKDOOR	1 746	583
	SHELLCODE	1 133	378
	WORMS	130	44
	TOTAL	175 341	82 332

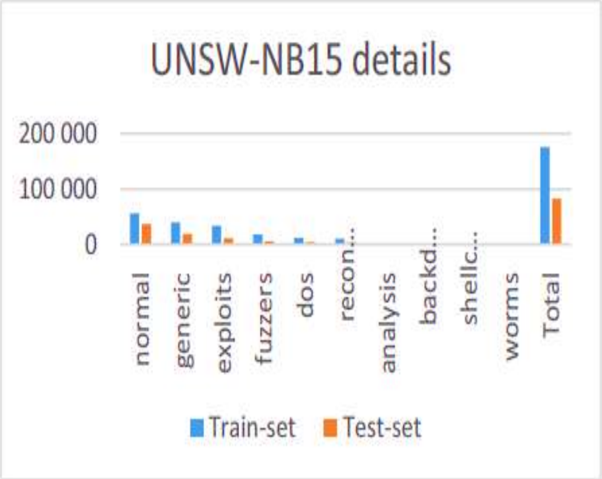


Fig. 1. Specifics of the UNSW-NB15

NSL-KDD

The KDD'99 dataset is currently out of date and contains duplicate records, making it unreliable for identifying network vulnerabilities. The issue has been resolved in NSL-KDD, an enhanced version of KDD'99. The NSL-KDD training set has 125,973 data points, whereas the testing set has 22,544 data points. The dataset has 41 variables, each with a label and data types including nominal, binary, and numeric values. The dataset includes four types of attacks: denial of service (DoS), probing, unauthorized remote access (R2L), and unauthorized local access (U2R). There is also a standard class category. Table 2 shows the distribution of each assault across the training and testing sets.

Table 2. The datasets of NSL-KDD.

Dataset	Class	Train-set	Test-set
NSL-KDD	normal	67 343	9,711
	dos	45 927	7 458
	probe	11 656	2 421
	r2l	995	2 754
	u2r	52	200
	Total	125 973	22 544

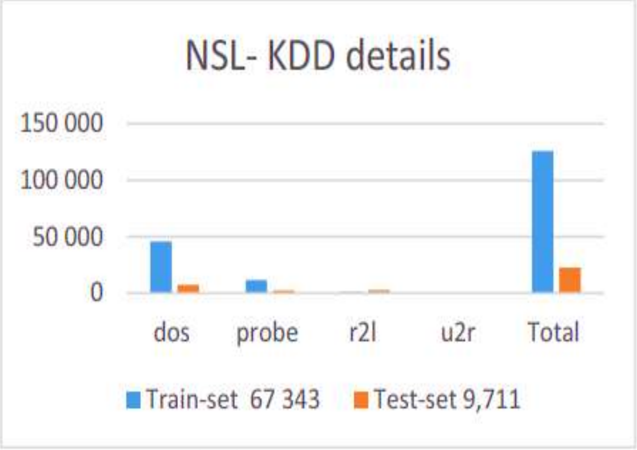


Fig. 2. Specifics regarding NSL-KDD

3. RELATED WORKS

Intrusion detection systems have been researched using a variety of approaches. The introduction of Big Data has presented problems to traditional data management strategies. As a result, several researchers want to construct an accurate and efficient intrusion detection system using big data approaches. This section focuses on academics who used machine learning approaches to tackle intrusion detection in the context of Big Data. Ferhat et al. used the cluster machine learning technique. To distinguish between a legitimate network traffic flow and a malicious attack, the authors used the k-Means technique from Spark's machine learning library. The KDD Cup 1999 dataset is used for both training and testing in the proposed methodology. The authors of this proposed approach identify the important features without using a feature selection algorithm.

Peng et al. proposed the use of principal component analysis (PCA) and the Mini Batch K-means clustering technique for intrusion detection systems. The dimension of the processed dataset is reduced by applying the principal component analysis technique, followed by data clustering with the micro batch K-means++ strategy. The proposed model has been tested using the entire KDDCup1999 dataset.

Peng et al. used machine learning to classify data. The authors suggested an Intrusion Detection System (IDS) specifically built for fog circumstances that uses decision trees to efficiently manage and evaluate massive amounts of data. The researchers' recommended methodology comprises adopting a preprocessing strategy to identify the strings in the given dataset, followed by data normalization to assure input data accuracy and increase detection efficacy. The decision tree methodology for IDS was compared to KNN and Naïve Bayesian methods. The experimental findings from the KDDCUP99 dataset demonstrated the proposed approach's effectiveness and accuracy.

Belouch et al. used Apache Spark to assess the effectiveness of SVM, Naïve Bayes, Decision Tree, and Random Forest classification algorithms for Intrusion Detection Systems (IDS). The UNSW-NB15 dataset is used to compare overall performance metrics such as prediction time, training time, and accuracy.

Furthermore, Manzoor and Morgan suggested a real-time intrusion detection system that uses the Apache Storm framework and Support Vector Machines (SVM). The authors used C-SVM and lib SVM for intrusion detection categorization. The suggested methodology was trained and tested using the KDD 99 dataset. Furthermore, many research used techniques for attribute selection. Vimalkumar and Randhika presented a Big Data system for detecting intrusions in smart grids. The framework uses many algorithms, including neural networks, SVMs, DTs, Naïve Bayes, and Random Forests. Furthermore, it employs the PCA Features selection technique. This method uses Principal Component Analysis (PCA) to reduce dimensionality and a correlation-based methodology to pick features. The proposed approach intended to improve classification accuracy while also shortening the time necessary for attack prediction. The Synchro Phasor dataset was used for both evaluation and training purposes. The proposed approach is evaluated using parameters such as specificity, recall, accuracy rate, and false positive rate (FPR).

Dahiya and Srivastava developed a Spark-based system for detecting intrusions with high efficiency and accuracy. The suggested system uses seven classification algorithms: Naïve Bayes, REP TREE, Random Tree, Random Forest, Random Committee, Bagging, and Randomizable Filtered. Feature reduction was also achieved using Canonical Correlation Analysis (CCA) and Linear

Discriminant Analysis (LDA). The proposed study used two sets of UNSW-NB 15 datasets to assess the efficacy of each classifier. The experimental results of the recommended technique showed that the LDA plus random tree algorithm strategy was more efficient and faster. The results show that the AUROC of dataset 1 was 99.1, while dataset 2 had an AUROC of 97.4. The AUROC value for our model was found to be 99.55. Consequently, our model is faster and more effective.

Hongbing Wang et al. developed the SP-PCA-SVM approach, which combines parallel principal component analysis (PCA) with parallel support vector machine (SVM) processing on the Spark platform. PCA is used for data analysis and feature extraction in order to reduce dimensionality using bagging. The KDD99 dataset was used in the suggested method for both training and evaluation.

Giura and Wang (2012) present a comprehensive and customizable strategy and structure for detecting Advanced Persistent Threats (APTs) that may be applied in any organizational situation. Beginning with the concept of an attack tree, they developed a conceptual attack structure known as the attack pyramid. The pyramid's apex represents the target of the attack, while the nearby planes represent potential attack sites. The habitats, or planes, are the user, application, network, physical, and others. The pyramid depicts the trajectory of the assault, beginning at the bottom and progressing upwards through the stages of data collection, operation, exploitation, delivery, reconnaissance, and exfiltration (with the objective serving as the ultimate goal). The enlarged attack pyramid displays potential pathways and vulnerabilities that span numerous layers with the goal of reaching the target.

Mirsky et al. (2018) offer a Network Intrusion Detection System (NIDS) that can be installed and used without any extra configuration. This Network Intrusion Detection System (NIDS) can efficiently detect and identify online and unmonitored attacks on a local network. The authors recognize that Kitsune seeks to address the limitations of previous approaches that use artificial neural networks (ANNs) as network intrusion detection systems. There are a few constraints to consider. First and foremost, a labeled dataset is necessary for successful deployment. Furthermore, most supervised learning algorithms can only address known threats and require a powerful CPU for model training. Furthermore, the newly trained models cannot be used until updates are sent to the organization's network intrusion detection system. Given these issues, Kitsune intends to act as an online, self-contained, machine learning-based lightweight Artificial Neural Network Intrusion Detection System (ANN-based NIDS). Its objective is to run in real time on network routers. The Kitsune framework consists of three major components: a feature extractor, a feature mapping, and a packet capturer and parser. The feature extractor is in charge of extracting data such as channel statistics. The feature mapper then combines these features into a bigger set that may be processed by the anomaly detector. Finally, the packet capturer and parser process data such as the packet's meta information. KitNET, Kitsune's principal anomaly detection method, uses a collection of miniature artificial neural networks called autoencoders to distinguish between normal and aberrant traffic patterns. The online feature extraction framework effectively monitors and analyzes all network channel trends. This framework makes use of damped incremental statistics stored in a hash table for this purpose.

Milajerdi et al. (2019) presented HOLMES as a tool for detecting actions associated with Advanced Persistent Threat (APT) campaigns. To generate a detection signal, their system collects warnings from a variety of sources within the company, including intrusion detection systems (IDSs) and audit data from hosts such as Linux auditd or Windows ETW. Broadly speaking, the methodologies they created take use of the linkages that exist between suspicious information flows, such as files and programs, that occur throughout a cyber attack operation. The system design enables the production of alarms that closely match the death chain, a series of procedures used by APT attackers. The cyber death chain represents the various stages of the APT lifecycle, beginning with initial observation and advancing to the ultimate goal of data exfiltration.

Nagaraja pioneered a new technique for detecting peer-to-peer (P2P) botnets. Botmasters started the spread of control throughout the network by using peer-to-peer (P2P) connections, which increased their capacity to endure disruptions and remain anonymous. To take use of this trait, the researchers use Markovian diffusion processes on network flows represented in a graph model. The P2P botnet graph topology is distinguishable from other network interactions. The fundamental premise is to identify patterns in traffic flows (how people communicate) and interconnection.

Oprea et al. (2015) created a technique for detecting early-stage Advanced Persistent Threats (APTs) based on belief propagation, as described by Yedidia et al. (2003). An Advanced Persistent Threat (APT) is a sophisticated and sustained cyberattack on a targeted entity. It frequently uses modern software and acts surreptitiously in order to blend in with its surroundings. The authors used the following common infection patterns to identify Advanced Persistent Threats (APTs): Connections with Command and Control (C&C) servers are established; HTTP(S) protocols are used to bypass firewalls; rapid visits to multiple domains are made, with redirection techniques used to conceal the attacker's identity; and domains linked to an attack are used, which share location/IP space, hosts, and access time. The goal was to identify suspicious communications coming from inside hosts, which are symptomatic of an Advanced Persistent Threat (APT).

4. CLASSIFICATION ALGORITHMS

The collected data can be useful for a variety of applications, including monitoring manufacturing operations, doing market research, and advancing scientific inquiries. Machine learning relies heavily on classification algorithms. Their duty is to divide unlabeled data into separate categories. The study uses the following algorithms:

Support vector machines (SVMs) are machine learning algorithms. The Support Vector Machine (SVM) technique provides a quick and simple prediction procedure, making it one of the most reliable machine learning classification algorithms when compared to others. The classification method uses support vectors from a data source to categorize data points based on a hyperplane.

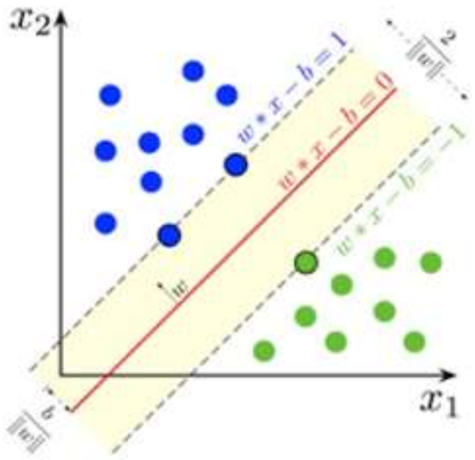


Fig. 3. SVM

K-Nearest Neighbor (KNN) is a reliable approach for categorizing data into different classes. One of its intriguing attributes is its applicability to both regression and classification tasks.

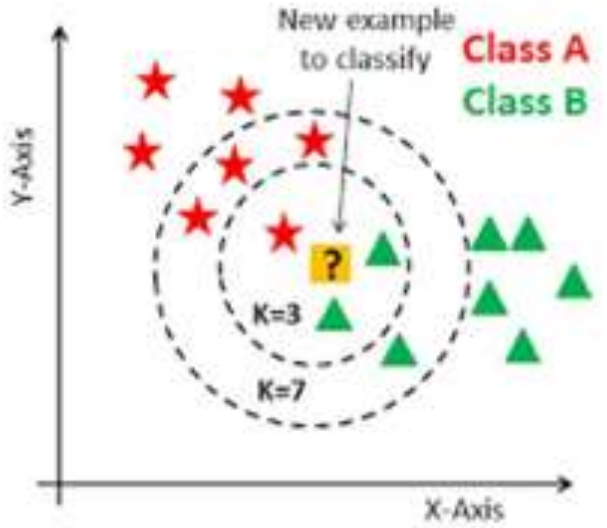


Fig. 4. Specifics regarding NSL-KDD

Naïve Bayes (NB) is a classification technique that estimates the likelihood of a given model belonging to a specific class. The concepts that underpin the Bayes theorem serve as its foundation. It is based on the idea that, for example, the attribute value for a certain class is unaffected by the attribute values. The concept is known as class conditional independence.

$$P(H/X) = P(X/H).P(H)/P(X) \qquad (1)$$

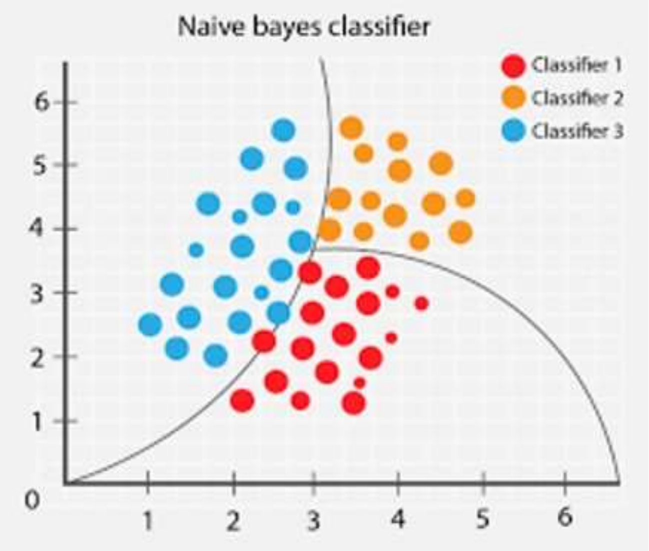


Fig. 5. Naïve Bayes

6. METHODOLOGY

For dataset classification, we contrasted the accuracy of SVM KNN with Naïve Bayes. We start with the raw dataset and sort the 19 different types of assaults into 5 categories using the class attribute. Normal, Dos, Probe, R2L, and U2R are the ones listed. Data points

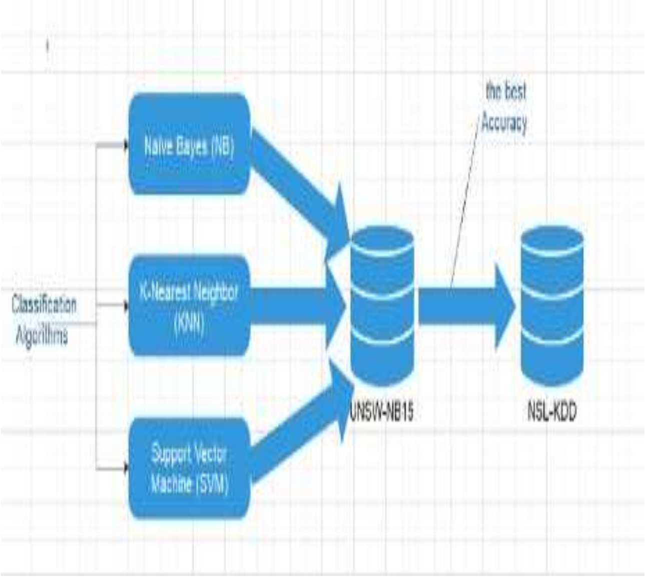


Fig. 6. Testing procedures and procedures

Table 3. Attacks' accuracy rate, as measured by the UNSW-NB15

	Accuracy
KNN (k=3)	93.3333%
NB	95.55555%
SVM	97.77777%

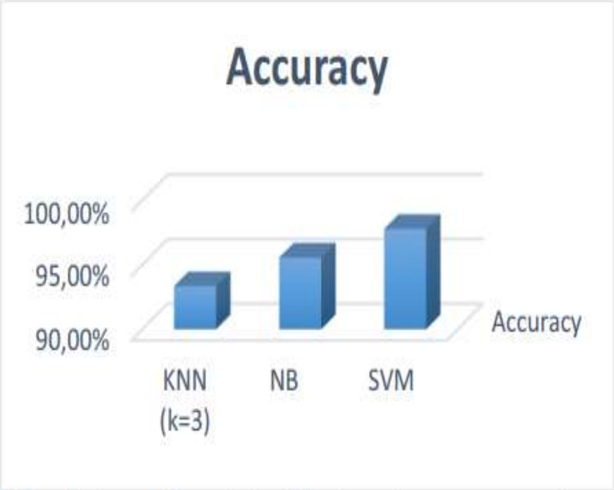
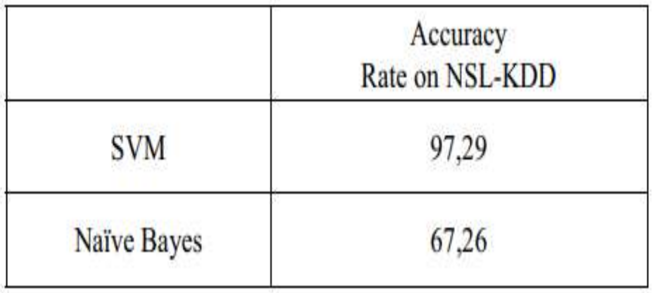


Fig. 7. A comparison of the performance of different classifiers on the UNSW-NB15

Table 4. (NSL-KDD) measures the accuracy rate of attacks.



	Accuracy Rate on NSL-KDD
SVM	97,29
Naïve Bayes	67,26

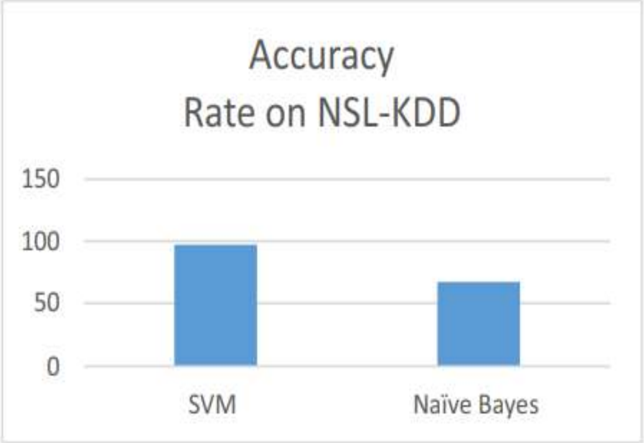


Fig. 8. A comparison of the performance of different classifiers on NSLKDD

The SVM algorithms demonstrated improved accuracy once more with a dataset of a similar sort, despite the fact that they are of different sizes and have different attacks.

7. CONCLUSION

The first database in this search is handled by three algorithms: Support Vector Machines (SVM), Naive Bayes (NB), and K-Nearest Neighbors (KNN), each with a neighborhood size of three. Following the acquisition of a substandard result, the KNN method is abandoned, and the remaining two algorithms are used to process the secondary database. The Support Vector Machine (SVM) has continuously demonstrated strong performance, regardless of the database's size or the type of the threats encountered. The following projects will work on improving this model's processing time. In addition, attempts will be made to link it with a firewall system and conduct real-time testing.

REFERENCES

1. Tchakoucht TA, Ezziyyani M. Building a fast intrusion detection system for high-speed-networks: probe and DoS attacks detection. Procedia Comput Sci. 2018;127:521–30.
2. M. Belouch, S. El Hadaj, and M. Idhammad. A two-stage classifier approach using reptree algorithm for network intrusion detection. International Journal of Advanced Computer Science and Applications, 8(6), pp.389- 394 (2017)
3. Zuech R, Khoshgoftaar TM, Wald R. Intrusion detection and big heterogeneous data: a survey. J Big Data. 2015;2:3.
4. M. Belouch, S. El Hadaj, & M. Idhammad. Performance evaluation of intrusion detection based on machine learning using Apache Spark. Procedia Computer Science, 127, 1-6,(2018).
5. Sahasrabuddhe A, et al. Survey on intrusion detection system using data mining techniques. Int Res J Eng Technol. 2017;4(5):1780–4.
6. N. Moustafa, N. (2017). Designing an online and reliable statistical anomaly detection framework for dealing with large high-speed network traffic (Doctoral dissertation, University of New South Wales, Canberra, Australia). (2017)
7. Dali L, et al. A survey of intrusion detection system. In: 2nd world symposium on web applications and networking (WSWAN). Piscataway: IEEE; 2015. p. 1–6.
8. W. Richert, L. P. Coelho, “Building Machine Learning Systems with Python”, Packt Publishing Ltd., ISBN 978-1-78216-140-0
9. M. Bkassiny, Y. Li, and S. K. Jayaweera, “A survey on machine learning techniques in cognitive radios,” IEEE Communications Surveys & Tutorials, vol. 15, no. 3, pp. 1136–1159, 2012.
10. Scarfone K, Mell P. Guide to intrusion detection and prevention systems (idps). NIST Spec Publ. 2007;2007(800):94. 6. Debar H. An introduction to intrusion-detection systems. In: Proceedings of Connect, 2000. 2000.
11. A. Iftikhar, M. Basheri, M. Javed Iqbal, A. Raheem; “Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection”, IEEE ACCESS, Survivability Strategies for Emerging Wireless Networks, 6 ,pp.33789-33795, (2018).