

Spark-Mllib intrusion detection mechanism using machine learning models

Asra Sarwath, Raafiya Gulmeher, Zeenath Sultana

Department of Computer Science and Engineering, Faculty of Engineering and Technology, Khaja Bandanawaz University (KBNU), Kalaburagi, India

Article Info

Article history:

Received Oct 14, 2023

Revised Nov 17, 2023

Accepted Nov 30, 2023

Keywords:

Classification

Deep learning

Intrusion detection system

Machine learning

PySpark

Spark Mllib

ABSTRACT

Typically, a single method is employed in machine learning (ML) based intrusion detection to identify intrusion information. However, this approach lacks flexibility, has a low detection rate, and struggles to handle high-dimensional data. Consequently, it is not efficient in addressing these challenges. This study proposes a new intrusion detection architecture that utilizes Spark and ensures resilient data dissemination across the platform to improve its effectiveness. It consists of preprocessing module, a label encoder module, a feature extraction module, a classification module and a database module. The preprocessing module compresses information by utilizing the module for label encoding. This generates a lower-dimensional reconstruction and classification characteristic. The database module has the capability to store the compressed characteristics of all traffic. This enables the classifier to be tested and then returns these features back into the original traffic, facilitating retraining. In order to evaluate the efficacy of the framework, simulations were conducted using the CICIDS 2017 dataset to accurately replicate the network traffic. Based on the test findings, the accuracy of both multiclass and binary classification surpasses that of earlier studies. High precision was achieved for the traffic that was restored. The possible application of the proposed architecture for edge/fog networks is discussed in the conclusion.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Asra Sarwath

Department of Computer Science and Engineering, Faculty of Engineering and Technology

Khaja Bandanawaz University (KBNU)

Kalaburagi, Karnataka, India

Email: asra.sarwath2003@gmail.com

1. INTRODUCTION

The proliferation of internet users and the integration of devices into the network are seeing a significant and swift growth. As the number of people and devices connecting to this network increases, the volume of data generated and stored is experiencing exponential growth. However, there has been a simultaneous increase in the number of cyber attacks [1] targeting the network and data storage. This results in the development of many techniques for identifying and responding promptly to unwanted actions and incidents occurring within the network. We introduce an intrusion detection system (IDS) [2] that incorporates a Spark resilient-inspired detector and response mechanisms. Due to the rapid and significant rise in the number of internet users, there has been a corresponding increase in the frequency of targeted attacks on the internet. Most attacks occur by exploiting vulnerabilities in unpatched programs of the operating system. According to the report [3], the prevalence of malware rose from 1 in 244 in 2014 to 1 in 131 by 2020. The national vulnerability database contains statistics on software vulnerabilities [4].

The programs with the most vulnerabilities are stated in [5]. Furthermore, there was a substantial rise in email malware, in addition to the vulnerabilities present throughout the entire system. Hence, the detection and response engines possess the capability to identify the constituents of networks.

Cyber attacks are crucial in causing instability and posing risks for the economic stability of a nation. According to the economic times [6], there was a greater than 20% loss in revenue in India during the year 2016. A total of \$209 million was repaid to ransomware hackers at the beginning of 2016 [7]. So, from a financial viewpoint, it is essential that the network, as well as the data within it, be protected with the least chance of being compromised. This implies the use of the detection and response system for the network. The majority of the latest research in IDS [8] is constructed using machine learning (ML) and other soft computing methods [9]. These systems require a lot of computational power, and their analysis can be slow; consequently, they are rarely applied to a real network. It is therefore difficult to identify intrusions in real-time. A system that is based on learning can be influenced by external influences from an attacker.

The most recent findings from research have identified the following issues to be further considered for investigation:

- Do learning-based IDS remain safe? Does it have any chance that an attacker could affect the external learning behavior that an IDS is able to perform? Does the result of classification by the IDS be affected by the course of time?
- Could attack detection be modeled in a more efficient way to create a beautiful and efficient system by drawing an inspiration from nature? Could new detection systems be quickly integrated in the overall system? Can the system be scalable?
- What are the potential living organisms in the world that can be considered for designing an IDS's response system? Can the approach be used to enhance performance of the IDS response systems?
- Can we achieve the retrieval of data from an attack on a network?

The paper's contribution is to examine the sturdiness of ML against manipulated data, and then to explore the possibilities of designing a system for intrusion detection as well as an attack response system that can be capable of surviving various attacks, and also perform a reliable backing up and recovering data that can be scalable and be executed without human intervention:

- To determine the possibility that a ML-based IDS is able to fail, you can use poisoned instances.
- In order to design a SparkML method of responding to an IDS.
- To design a vector assembler method called chi2 extratree classifier to extract the characteristics from data and data recovery, as an extended response to an IDS developed for the second goal.
- Analyze the proposed technique in different scenarios of an adversarial attack.

2. RELATED WORK

Al-Janabi and Saeed [10] proposed an intrusion detection model employing BPANN to distinguish abnormal network traffic from normal traffic. It achieved a precision of around 93%. Meng [11] the algorithm was suggested as an anomaly detection technique that was based on the algorithm of decision trees and compared its performance against artificial neural networks and support vector machines, as well as other methods of decision trees. After analysis of the results, it was concluded that the proposed method is superior to others. It was also discovered that the detection rates of attacks using R2L and U2R are high, while the detection rates of the most frequent attacks, such as denial of service (DoS) and probing attack (Probe), are very low. Therefore, this algorithm is only suitable to detect low-frequency attacks. Gadal *et al.* [12] suggested an anomaly detection model, which is hybrid since it incorporates the clustering process and also methods for classification. In order to cluster, k-mean clustering techniques are utilized, as well as sequential minimal optimization (SMO) to classify. The analysis results are better when it is a hybrid.

Ibrahim *et al.* [13] a model for intrusion detection was proposed that is based on methods of ML like C5, multilayer perception (MLP), and Naive Bayes (NB). By using these ML techniques, a multilevel classification tool is created. In each level, various techniques for classification are used, and at each level, only one attack is classified. The analysis results show that multilevel methods have greater accuracy in detecting than one technique. Numerous ML techniques can be used to decrease the false alarm rate for IDS based on anomalies.

Feng *et al.* [14] utilized extreme learning machines (ELMs) together with models that combine several other techniques for ML. Each model has distinct strengths and weaknesses, and general detection rates are growing. In comparison to other models, ELM equipped with support vector machine (SVM) surpasses the good detection rate in separating information flow in networks as normal and abnormal packets.

Hornig *et al.* [15] a system for intrusion detection was suggested that is based on clustering and classification methods. The BIRCH algorithm is used for clustering, and for clustering SVMs are used. The results of the analysis show a higher accuracy of around 96%, and the false alarm rate is 0.7%.

Research has proposed many methods for IDS working with ML [16] and deep learning [17] models with knowledge discovery databases (KDD) Cup 1999 and CICIDS 2017 datasets. Recently, as the data has grown to massive amounts, big data techniques have been introduced to work on the intrusion with massive amounts of data. The authors implemented the models with Spark-enabled methods. The research work needs to progress in Spark ML with classification algorithms using Pyspark for efficient results. With the challenge of large-volume and high-dimensional data, though there are many ways to use unsupervised learning to detect network intrusions that have been proposed in recent times, some are still prone to weaknesses and issues. The predictions derived from high-dimensional learning contain duplicate features, which can reduce the accuracy of classifications in the raw dataset.

2.1. Novel perspective

We are committed to develop an efficient method for generalization and classification. The main challenge today is to protect users from the security threats [18] on internet. IDS are among the security tools readily available to detect potential attackers on networks or hosts. The model has been created by using the robust data distribution technique of Spark and similar methods of ML with the chi2 extra tree classifier, which allows feature extraction using the vector assembler method for performance. The contamination is caused by anomaly detection by intelligent IDS using the label encoder construction error as a metric to measure anomaly discovery. It uses the NSL-KDD [19] to evaluate the effectiveness of the model by using both test and training temporal metrics datasets to evaluate performance, as well as maintaining an unchanging detection environment and reducing the level of contamination in the training data.

3. PROPOSED METHOD

The proposed framework is discussed in this article. The IDS is composed of the following elements, preprocessing module [20], label encoder module, feature extraction module, classification modules including database module. They are all kept in order to create an effective intrusion detection framework. It is extremely accurate and has minimal training complexity.

3.1. Workflow

We will discuss each module of the framework individually with the role;

- Preprocessing module: analyzed in a specified method to get the raw data.
- Label encoder module: the module is made up of a feature selection model that analyzes the data, eliminates the features that are not important to the dataset, and extracts the low-dimensional features that are reconstructed.
- Feature extraction module: this module mostly employs supervised algorithms to classify elements, detect pattern that could trigger attacks, and decide whether or not a warning is issued in response to the findings.
- Classification module: classify traffic, decide if an attack is taking place, and whether or not a warning is issued according to the outcomes.

3.2. Classification models

The classification module employed random forest (RF) [21], [22] for its primary algorithm. In order to evaluate the effectiveness of RF, we used decision trees (DT) in the initial stage. The lower the value, you will get more attributes and therefore more purity. Following division of this set, we choose the sub attribute with the lowest Gini index [23]. Classification and regression trees (CART) make its Gini coefficient lower with each iteration as shown in (1) and (2).

$$Gini(D) = 1 - \sum_{k=1}^K \left(\frac{|C_k|}{|D|} \right)^2 \quad (1)$$

$$Gini(D, A) = \sum_{i=1}^n \frac{|D_i|}{|D|} Gini(D_i) \quad (2)$$

4. EXPERIMENTAL RESULTS

4.1. Attack class

Probe: an attacker may be able to scan networks to obtain information and vulnerabilities. Security measures are scanned and abused using a network map. Attacks on probes can compromise the computer's functioning features. Probe attacks are one of the top attacks, e.g., portsweep, ipsweep, Nmap, and the Satan [24]. DoS: is an attack in the course of which an adversary sends an avalanche of requests for traffic [25] at a system in order to create the memory or computing resource that is too busy or full to manage legitimate

requests. In the process, it blocks authentic customer access to a PC. Bouyeddou *et al.* [26] is a DoS attack that targets computer such as Neptune, Neptune’s back pod, teardrop or land. CICIDS 2017 dataset [27]: the majority of current datasets are insufficient and outdated (some frequently used to evaluate intrusion detection include KDD’99 [28] as well as KDD-NSL). Some of them aren’t as diverse and have a large volume of data to encompass a variety of known threats, while others hide the payload of packets and don’t reflect current trends. Figure 1 illustrates the workflow of proposed model.

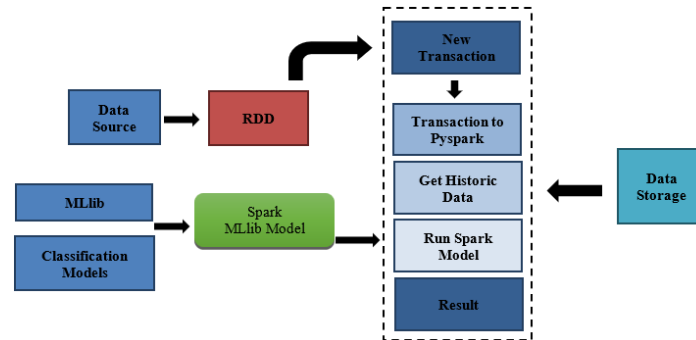


Figure 1. Workflow model

To prepare the dataset for processing, the NSL-KDD dataset is used, which comes from KDD’99. The dataset comprises two parts: the training set as and the testing set. First, we need to allocate column names for the set, and then look them up. This can help us identify the steps to be taken prior to processing for data to ensure it is suitable to be used for training. Each attack is recorded for the training sets and the percentage of each class being consistent with its type is noted. A similar procedure is used in the data for testing. Figure 2 represents the dataset with feature and target variables required for testing. The next step is to scale the attributes before encoding them. Figure 3 describes the port connectivity using Spark session builder for better performance.

	0	1	2	3	4	0.1	0.2	0.3	0.4	0.5	0.6	...	10.1	0.04.1	0.06.1	0.00.3	0.00.4	0.00.5	0.00.6	1.00.2	1.00.3	anomaly
0	0	tcp	private	REJ		0	0	0	0	0	0	...	1	0.00	0.06	0.00	0.00	0.00	0.00	1.00	1.00	anomaly
1	2	tcp	ftp_data	SF	12983	0	0	0	0	0	0	...	86	0.61	0.04	0.61	0.02	0.00	0.00	0.00	0.00	normal
2	0	icmp	eco_i	SF	20	0	0	0	0	0	0	...	57	1.00	0.00	1.00	0.28	0.00	0.00	0.00	0.00	anomaly
3	1	tcp	telnet	RSTO	0	15	0	0	0	0	0	...	86	0.31	0.17	0.03	0.02	0.00	0.00	0.83	0.71	anomaly
4	0	tcp	http	SF	267	14515	0	0	0	0	0	...	255	1.00	0.00	0.01	0.03	0.01	0.00	0.00	0.00	normal

Figure 2. Dataset with feature and target variables

```

] from pyspark.sql import SparkSession

] spark = SparkSession.builder\
    .master("local")\
    .appName("Colab")\
    .config('spark.ui.port', '4050')\
    .getOrCreate()

] spark

SparkSession - in-memory
SparkContext
Spark UI
Version
v3.2.1
Master
local
AppName
  
```

Figure 3. Spark session builder with port connectivity

4.2. Encoding of categorical attributes

To encode categorical attributes, all text information in the datasets illustrated in Figure 4 is converted into numerical data represented in Figure 5 followed by Figure 6 which shows preprocessing of

data with best features. The data is recognized by many models which eliminates the features that are not important using label encoder. Encoded text is replaced with column values that contain categorical text. A class column can be created by inserting the encoded text for the class. It is used for building data for different classifiers.

real	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	real1	hot	num_failed_logins	logged_in	num_compromised	root_
0	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0
1	2	tcp	ftp_data	SF	12983	0	0	0	0	0	0	0	0	0
2	0	icmp	eco_i	SF	20	0	0	0	0	0	0	0	0	0
3	1	tcp	telnet	RSTO	0	15	0	0	0	0	0	0	0	0
4	0	tcp	http	SF	267	14515	0	0	0	0	1	0	0	0

Figure 4. Dataset with object categorical columns

real	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	real1	hot	num_failed_logins	logged_in	num_compromised	root_
0	0	1	45	1	0	0	0	0	0	0	0	0	0	0
1	2	1	19	9	12983	0	0	0	0	0	0	0	0	0
2	0	0	13	9	20	0	0	0	0	0	0	0	0	0
3	1	1	55	2	0	15	0	0	0	0	0	0	0	0
4	0	1	22	9	267	14515	0	0	0	0	1	0	0	0

Figure 5. Label encoder applied on the dataset parameters converted to numeric elements

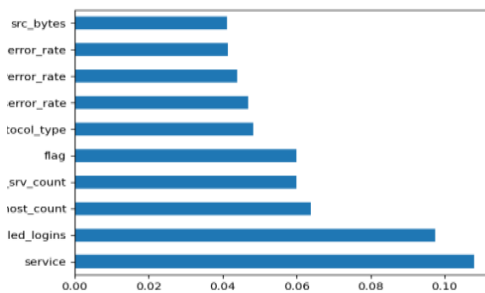


Figure 6. Pre processing of data with best features

4.2.1. Feature selection

A technique called feature selection is utilized. ML is a method of reducing the number and size of features made possible by predictive analytics. There are many options for selecting features, the most pertinent feature in selecting data is shown in Figure 7. In this study, the wrapper technique [29] will be used for feature selection [30] as illustrated in Figure 7(a), and complete dataset is been break down to feature variables and targeted variables as shown in Figure 7(b).

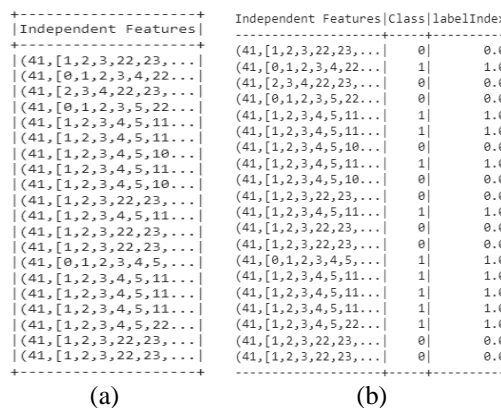


Figure 7. The most pertinent feature in selecting data (a) spark enabled feature and (b) feature variables and targeted variables

4.2.2. Building models

The models are built using classification algorithms. Then, we present the robust Spark model using PySpark [31], [32] that yields accurate results when implementing the vector assembler techniques that create feature variables into one list, with all parameters in only one column. The column that is targeted as a separate list is then stored as a two-column dataset. The machine library MLlib is implemented, and the corresponding accuracy and precision are determined.

4.2.3. Results and analysis

A comprehensive study was carried out based on the efficiency of several classifier models using Spark MLlib [33], [34]. A variety of evaluation parameters have been analyzed to determine the best one. The Figure 8 demonstrates the various feature variables with different scores that are generated using chi-square model. Figure 9 depicts performance of different ML models with respect to different parameters. Figure 9(a) represents different values of accuracy and precision of ML models under Spark resilient distribution and Figure 9(b) shows the sensitivity and specificity of different ML models.

Specs	Score
src_bytes	1.015220e+08
dst_bytes	3.690576e+07
real	2.300625e+06
dst_host_count	7.044734e+05
is_guest_login	3.725777e+05
srv_diff_host_rate	1.272281e+05
count	5.152715e+04
service	2.476673e+04
flag	1.060705e+04
num_failed_logins	3.819467e+03
srv_error_rate	3.787970e+03
error_rate	3.738992e+03
dst_host_error_rate	3.569640e+03
dst_host_srv_error_rate	3.451346e+03
dst_host_srv_count	2.328061e+03
error_rate	1.663790e+03
srv_count	1.649107e+03
dst_host_error_rate	1.624100e+03
dst_host_srv_diff_host_rate	1.534938e+03
srv_error_rate	1.350584e+03
su_attempted	1.184360e+03
same_srv_rate	1.076657e+03
dst_host_same_srv_rate	7.812280e+02
logged_in	6.845998e+02

Figure 8. Feature variables with different scores generated from chi-square models

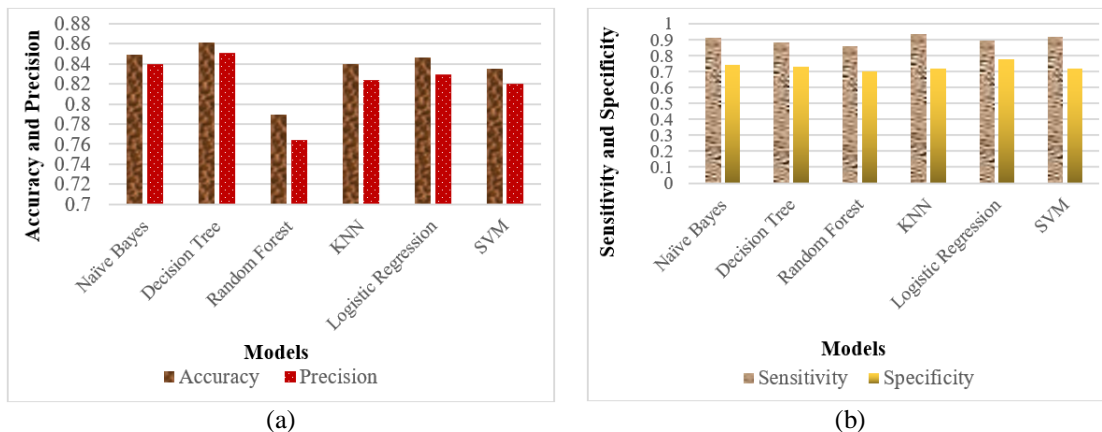


Figure 9. Depicts performance of different ML models with respect to different parameters (a) accuracy and precision results of ML models and (b) sensitivity and specificity results from the models

5. CONCLUSION

The KDD-Cup99 dataset was once considered the standard dataset for intrusion detection. However, due to significant changes in both network architecture and attack patterns, the CICIDS 2017 dataset has now emerged as the new benchmark. Hence, it can be employed to identify attacks that exploit the existing

network conditions. The utilization of Spark MLlib, a technology that employs distributed data processing and is resilient to errors, produced effective outcomes in identifying abnormalities. The experimental results demonstrate that the suggested approach is capable of constructing an intrusion detection model that exhibits a high detection rate, high precision, and a low false positive rate. Subsequently, the following action is extracting real-time data packets from the network and subjecting them to examination using the preclassified training data. Given the acquired results, this research has the potential to be extended to include host-based IDS or analysis at a specific application layer.




REFERENCES

- [1] P. Nayak, M. Sufiyan, Monisha N S, M. G. Bhaskar, and M. Raju, "Review paper on cyber security and types of cyber attacks," *International Journal of Advanced Research in Science, Communication and Technology*, pp. 732–735, Aug. 2022, doi: 10.48175/ijarsct-7043.
- [2] M. Pradhan, C. K. Nayak, and S. K. Pradhan, "Intrusion detection system (IDS) and their types," in *Securing the Internet of Things: Concepts, Methodologies, Tools, and Applications*, IGI Global, 2019, pp. 481–497.
- [3] "Internet Security Threat Report 2017," 2017. <https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf> (accessed Jun. 20, 2017).
- [4] X. Li, X. Chang, J. A. Board, and K. S. Trivedi, "A novel approach for software vulnerability classification," in *Proceedings - Annual Reliability and Maintainability Symposium*, 2017, pp. 1–7, doi: 10.1109/RAM.2017.7889792.
- [5] Calyptix Security, "Top security threats 2017 what's ahead and how to prepare," 2017. <http://www.calyptix.com/wp-content/uploads/Top-Threats-2017-Report.pdf> (accessed Jun. 20, 2017).
- [6] E. Times, "Cyber attack caused over 20% revenue loss in 2016," 2017, <https://www.pressreader.com/india/economic-times/20170207/281814283602169>, Feb. 2017. [Online; accessed 20-June-2017].
- [7] McAfee, "McAfee labs," 2014. <https://www.mcafee.com/in/resources/reports/rp-threats-predictions-2017.pdf> (accessed Jun. 20, 2017).
- [8] V. Manekar and K. Waghmare, "Intrusion detection system using support vector machine (SVM) and particle swarm optimization (PSO)," *International Journal of Advanced Computer Research*, no. 3, pp. 2–6, 2014.
- [9] M. Murakami and N. Honda, "Performance of the IDS method as a soft computing tool," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 6, pp. 1582–1596, Dec. 2008, doi: 10.1109/TFUZZ.2008.2005693.
- [10] S. T. F. Al-Janabi and H. A. Saeed, "A neural network based anomaly intrusion detection system," in *Proceedings - 4th International Conference on Developments in eSystems Engineering, DeSE 2011*, Dec. 2011, pp. 221–226, doi: 10.1109/DeSE.2011.19.
- [11] Y. X. Meng, "The practice on using machine learning for network anomaly intrusion detection," *Proceedings - International Conference on Machine Learning and Cybernetics*, vol. 2, pp. 576–581, 2011, doi: 10.1109/ICMLC.2011.6016798.
- [12] S. Gadal, R. Mokhtar, M. Abdelhaq, R. Alsaqour, E. S. Ali, and R. Saeed, "Machine learning-based anomaly detection using k-mean array and sequential minimal optimization," *Electronics (Switzerland)*, vol. 11, no. 14, p. 2158, Jul. 2022, doi: 10.3390/electronics11142158.
- [13] H. E. Ibrahim, S. M. Badr, and M. A. Shaheen, "Adaptive layered approach using machine learning techniques with gain ratio for intrusion detection systems," *International Journal of Computer Applications*, vol. 56, no. 7, pp. 10–16, Oct. 2012, doi: 10.5120/8901-2928.
- [14] W. Feng *et al.*, "Mining network data for intrusion detection through combining SVMs with ant colony networks," *Future Generation Computer Systems, Elsevier*, Vol. 37, 2014, pp. 127–140, doi: <https://doi.org/10.1016/j.procs.2017.12.204>.
- [15] S. J. Horng *et al.*, "A novel intrusion detection system based on hierarchical clustering and support vector machines," *Expert Systems with Applications*, vol. 38, no. 1, pp. 306–313, Jan. 2011, doi: 10.1016/j.eswa.2010.06.066.
- [16] A. S. Jaradat, M. M. Barhoush, and R. B. Easa, "Network intrusion detection system: Machine learning approach," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 2, pp. 1151–1158, Feb. 2022, doi: 10.11591/ijeecs.v25.i2.pp1151-1158.
- [17] W. G. Hatcher and W. Yu, "A survey of deep learning: platforms, applications and emerging research trends," *IEEE Access*, vol. 6, pp. 24411–24432, 2018, doi: 10.1109/ACCESS.2018.2830661.
- [18] M. Abomhara and G. M. Køien, "Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks," *Journal of Cyber Security and Mobility*, vol. 4, no. 1, pp. 65–88, 2015, doi: 10.13052/jcsm2245-1439.414.
- [19] L. Dhanabal and S. P. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 6, pp. 446–452, 2015, doi: 10.17148/IJARCC.2015.4696.
- [20] T. Ahmad and M. N. Aziz, "Data preprocessing and feature selection for machine learning intrusion detection systems," *ICIC Express Letters*, vol. 13, no. 2, pp. 93–101, 2019, doi: 10.24507/icicel.13.02.93.
- [21] S. Chimphee and W. Chimphee, "Machine learning to improve the performance of anomaly-based network intrusion detection in big data," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 30, no. 2, p. 1106, May 2023, doi: 10.11591/ijeecs.v30.i2.pp1106-1119.
- [22] J. Zeffora and Shobarani, "Optimizing random forest classifier with Genesis-index on an imbalanced dataset," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 26, no. 1, pp. 505–511, Apr. 2022, doi: 10.11591/ijeecs.v26.i1.pp505-511.
- [23] S. Tangirala, "Evaluating the impact of GINI index and information gain on classification using decision tree classifier algorithm," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 2, pp. 612–619, 2020, doi: 10.14569/ijacsa.2020.0110277.
- [24] B. Kaur, "Classification of attacks in data mining," *International Journal of Innovations in Engineering and Technology*, vol. 8, no. 1, 2017, doi: 10.21172/ijiet.81.044.
- [25] M. Alizadeh, M. T. H. Beheshti, A. Ramezani, and H. Saadatinezhad, "Network traffic forecasting based on fixed telecommunication data using deep learning," in *6th Iranian Conference on Signal Processing and Intelligent Systems, ICSPIS 2020*, Dec. 2020, pp. 1–7, doi: 10.1109/ICSPIS51611.2020.9349573.
- [26] B. Bouyeddou, F. Harrou, Y. Sun, and B. Kadri, "Detection of smurf flooding attacks using Kullback-Leibler-based scheme," in *2018 4th International Conference on Computer and Technology Applications, ICCTA 2018*, May 2018, pp. 11–15, doi: 10.1109/CATA.2018.8398647.




- [27] A. Boukhamla and J. C. Gavro, "CICIDS2017 dataset: performance improvements and validation as a robust intrusion detection system testbed," *International Journal of Information and Computer Security*, vol. 16, no. 1–2, pp. 20–32, 2021, doi: 10.1504/IJICS.2021.117392.
- [28] S. Choudhary and N. Kesswani, "Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT," *Procedia Computer Science*, vol. 167, pp. 1561–1573, 2020, doi: 10.1016/j.procs.2020.03.367.
- [29] J. Maldonado, M. C. Riff, and B. Neveu, "A review of recent approaches on wrapper feature selection for intrusion detection," *Expert Systems with Applications*, vol. 198, p. 116822, Jul. 2022, doi: 10.1016/j.eswa.2022.116822.
- [30] G. V. Gopal and G. R. M. Babu, "An ensemble feature selection approach using hybrid kernel based SVM for network intrusion detection system," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 1, p. 558, 2021, doi: 10.11591/ijeecs.v23.i1.pp558-565.
- [31] R. Bandi, J. Amudhavel, and R. Karthik, "Machine learning with PySpark – review," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, no. 1, pp. 102–106, Oct. 2018, doi: 10.11591/ijeecs.v12.i1.pp102-106.
- [32] S. Selvakani, K. Vasumathi, and M. Ajith, "Predicting the crimes based on the weather using pyspark," *International Journal of Engineering and Advanced Technology*, vol. 8, no. 6, pp. 885–890, Aug. 2019, doi: 10.35940/ijeat.F8226.088619.
- [33] D. Sisiaridis and O. Markowitch, "Feature extraction and feature selection: reducing data complexity with apache spark," *International Journal of Network Security & Its Applications*, vol. 9, no. 6, pp. 39–51, 2017, doi: 10.5121/ijnsa.2017.9604.
- [34] N. Deshai, B. V. D. S. Sekhar, and S. Venkataramana, "Mllib: machine learning in apache spark," *International Journal of Recent Technology and Engineering*, vol. 8, no. 1, pp. 45–49, 2019, doi: A10090681S319/19©BEIESP.

BIOGRAPHIES OF AUTHORS






Asra Sarwath    perceived B.E., M.Tech., from Visvesvaraya Technological University 2007 and 2014 respectively. She has teaching experience of 15+ years and is currently working as an Assistant Professor in the Department of Computer Science and Engineering, Faculty of Engineering and Technology, KBNU, Karnataka, India. She is the research scholar at KBNU, Kalaburagi, Karnataka, India. Her area of interest in research work is deep learning, machine learning, cyber security, IoT. She has published many articles in national and international journals. She can be contacted at email: asra.sarwath2003@gmail.com.



Dr. Raafiya Gulmeher    is currently working as an Assistant Professor in the Department of Computer Science and Engineering, Faculty of Engineering and Technology, Khaja Bandanawaz University, Kalaburagi, Karnataka, India. She received Ph.D. degree in Computer science from JJTU, Rajasthan, India. She has more than 20 years of academic experience and have published many articles in international and national journals. She can be contacted at email: profraafiya.cse@gmail.com.



Zeenath Sultana    perceived B.E., M.Tech., from Visvesvaraya Technological University 2006 and 2010 respectively. She has teaching experience of 15+ years and is currently working as an Assistant Professor in the Department of Computer Science and Engineering, Faculty of Engineering and Technology, KBNU, Kalaburagi, Karnataka, India. She is the research scholar at KBNU, Kalaburagi, Karnataka, India. Her area of interest in research work is deep learning, machine learning, cloud computing, cyber security. She has published many articles in national and international journals and is an active reviewer for numerous international journals. She can be contacted at email: profzeenathcse@gmail.com.