Strategic Framework for Network Anomaly Detection Using Hybrid Machine Learning

Rejeti Kartik
Department of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
bl.en.u4cse21170@bl.students.amrita.edu

Anirudh S.
Department of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
bl.en.u4cse21020@bl.students.amrita.edu

P. Radha Nishant
Department of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
bl.en.u4cse21161@bl.students.amrita.edu

Sanjay Baitha
Department of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
bl.en.u4cse21185@bl.students.amrita.edu

Shinu M. Rajagopal*
Department of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
mr_shinu@blr.amrita.edu

Sreebha Bhaskaran
Department of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
b_sreebha@blr.amrita.edu

***Abstract*—In the rapidly evolving digital landscape network security and efficiency are important especially with the increasingly widespread network anomalies that can cause severe performance degradation. This study introduces a hybrid machine learning approach that combines Random Forest and Naive Bayes algorithm to address anomaly detection challenges. Using the 'UNSW-NB15_1' dataset, data preprocessing is performed, which includes removing missing values and standardizing attributes. The hybrid model uses random forest to select features and identify key features which is then used by Naïve Bayes Classifier for effective anomaly detection This method not only accurately predicts anomalies but also provides network weakness insights contributing to more secure network connection.**

***Index Terms*—Anomalies, Packet traffic, Machine Learning, Naïve Bayes, Random Forests**

I. Introduction

Machine Learning is a branch of computers, under Artificial Learning, which deals with how to teach a system or machine to think like a human and try to predict the outcome of certain data or information using certain algorithms. Machine Learning can be classified into supervised and unsupervised learning. Supervised learning is where a set of algorithms is used, to try to predict outcomes or classify the data into different classes. Unsupervised learning is where the user has to decide where the boundaries of the outcome should lie on. Supervised learning is further broken down into classification and regression. Classification is a method to separate the data into different groups or classes like they are separate boxes. Showing that a network is functional or not, is an example of classification. Regression is a method to find a correlation between different attributes, and get a relation that can be represented on a plot to graph. Locating packets in the system at a certain period of time is an example of regression. Classification determines discrete values, whereas regression determines continuous values. This would show an insight on how it will be useful for applications in networks.

Network anomalies are network outliers that are hostile in nature and disrupt in various ways with system functionality [1]. Some of these anomalies can be the result of an unanticipated packet loss; while others include contagious data. These can occasionally render a switch or router useless, rerouting the transfer into an alternative path and increasing the delay. At other times, they may overwhelm the system if it receives an excessive amount of requests at once. Sometimes, they may be of a fictitious IP address. A technique called network anomaly detection, is used to identify these erroneous packets or requests and flag them for attention by the network [2]. It can determined whether or not the data being transmitted is anomalous, by applying machine learning. Before the data reaches the host or router, this enables people to gain an understanding of it. This information can be shared with the network, so it may take appropriate action.

This study will work on the following

- The proposed method to obtain a high amount of accuracy in detecting anomalous packets from benign packets better compared to other models, as well as performing well in other metrics.

*Corresponding author.

- The time for the model to detect the packet when it is entering the system is ought to be faster than other models.

This study is divided into multiple sections. Section I which is being discussed currently, is about the introduction of what network anomalies are and what could be the solution for it. Section II talks about the literature survey. Section III talks about the proposed methodology. Section IV talks about the results obtained by the proposed method and compared with baseline models. Section V talks about the advantages, drawbacks and the future work that can be done.

II. Literature Review

The following is a discussion about the studies done by other researchers on the topic of Network Anomaly detection. They discuss the various methods to identify anomalies. Vijaya et al. discuss on how to identify the most popular applications and usage patterns to detect anomalies [3]. The study uses the Random Forest and Convolutional Neural Network to classify the packets and Wireshark for visualization on the 'Canadian Institute of Cybersecurity ISC VPN 2016' dataset. The accuracy it receives is 88% and precision identifies up to 99%. The proposed solution method would require more training.

Satheesh et al. discuss about controlling the flow of the data packets in the network, monitoring for abnormal traffic flow to detect anomalies [4]. The study propose a Software defined Network (SDN) and comparing with K-NN classifier, Support Vector Machine (SVM), Naive Bayes classifier (NB) on the NSL-KDD dataset. The detection rate of the SDN method was 95.16% and the false positive rate was 2.49%. The K-NN method had a detection rate of 85.48% and a false positive rate of 2.77% . The SVM method had a detection rate of 85.16% and false positive rate of 3.32%. The NB method had a detection rate of 89.03% and a false positive rate of 1.39%. The proposed model performs better than other models, but is also more vulnerable to attacks due to a unified design.

Zhipeng et al. perform a comparative analysis on the effectiveness of various machine learning algorithms for detecting anomalies on IoT networks [5]. Applying Logistic regression (LR), Support Vector Machine (SVM), K - Nearest Neighbors classifier (k-NN), Random Forest (RF), and XGBoost model on the 'SKT NUGU' dataset. The RF approach gave the best accuracy of 100% and recall value of 100% but it took the longest time to compute. XG Boost gives an accuracy of 97% and recall of 96%, and its computational time was the least.

Aditya et al. propose a method for detecting the anomalous points or malicious attacks, and checking the effectiveness using unsupervised machine learning [6]. They compare between Isolation Forest classifier and one-class Support Vector Machine (SVM) on the 'KDD cup' and 'NSL-KDD' datasets. The AUC score is around 98.3%. The contamination parameter value is 4% of the total number of samples that is 0.04. The "n_estimator" parameter was kept at 100. Therefore Developing Intrusion Detection System(IDS) is highly efficient in detecting the real-time anomalies.

Yu et al. propose a complete study on methods of anomaly detection for industrial production products based on Deep learning [7]. They use ROI Classifier, Fast-AnoGAN, WGAN_GP model, AnoGAN model, YOLOv3 model and Tiny-YOLOv3 model and compares their strength on the dataset. Both the YOLOv3 and Tiny-VOLOv3 anomaly detection models got a detection rate of 100%. The FPS rate of YOLOv3 is 5.34 and that of Tiny-YOLOv3 is 22.25. The missing rate of YOLOv3 is 0.0%, whereas the missing rate of Tiny-YOLOv3 is 2.50%. In the case of AnoGAN, Fast-AnoGAN models, we got detection rate of 100%, with FPS rate of 0.97 and 10. The speed of Fast-AnoGAN is 0.10s, whereas the speed of AnoGAN is 1.03s.

Igor et al. enhance network anomaly detection using machine learning on NetFlow data, establishing benchmarks with 'UNSW-NB15' dataset and metrics like accuracy and Area Under Receiver Operating Characteristics (ROC) Curve (AUC) [8]. A comparative analysis is being applied with the Stochastic Gradient Descent (SGD) model, Support Vector Machine (SVM), K-Nearest Neighbor (K-NN) classifier, Naive Bayes (NB) classifier, Decision Tree (DT) classifier, Random Forest (RF) classifier, and AdaBoost (AB) model on the 'UNSW-NB15' dataset. The RF Classifier with an F2-score of 97.68% and an AUC score of 98.47% gives the best results using a representative subset of the original dataset.

Richa et al. explore network anomaly detection amid rising network traffic, comparing machine learning techniques and identifying research gaps [9]. It provides insights for future advancements in the field. The study considers supervised, unsupervised, and reinforcement Learning. It compares algorithms like k-Nearest Neighbors (k-NN) classifier, Decision Trees, Ensemble classifier, Naïve Bayes classifier, Support Vector Machine (SVM), and Artificial Neural Network (ANN) in supervised Learning, and explores Hidden Markov Models and clustering methods in unsupervised Learning. Reinforcement Learning also examines for both categorical and continuous data. It discusses various ML methods and their performance on datasets, emphasizing SVM's accuracy. The survey also addresses challenges in real-time anomaly detection, advocating for accuracy evaluation in dynamic environments, not just static datasets.

Tharindu et al. address cloud network security in complex cloud computing environments [10]. It employs machine learning, specifically one class Support Vector Machine and Autoencoder model, for anomaly detection in cloud network data. Using benchmark datasets, it shows neural network-based methods are more effective in detecting anomalies than kernel-based methods. The study emphasizes the significance of anomaly detection in maintaining the integrity and confidentiality of cloud systems. The study uses YAHOO Synthetic and 'UNSW-NB15' datasets for cloud network anomaly detection. one-class Support Vector Machine (OCSVM) achieves 79.17% accuracy on YAHOO data and 60.89% on 'UNSW-NB15'. Autoencoder, a neural network method, reaches 96.02% accuracy on YAHOO and 99.10% on 'UNSW-NB15'. The autoencoder model outperforms the OCSVM model, highlighting its effec-

TABLE I
RELATED WORKS ON NETWORK ANOMALY DETECTION

Reference No.	Model Used	Insights obtained
[3]	Random Forest and CNN	Accuracy of 88%
[4]	Software Defined Network	Detection rate of 95.16%
[5]	Random Forest	Accuracy of 100%
[6]	Unsupervised Machine Learning	AUC score of 98.3%
[7]	Tiny YOLOv3 model	Detection rate of 100%
[8]	Random Forest	AUC score of 98.47%
[9]	Support Vector Machine	Accuracy obtained is high
[10]	Auto-encoder neural network	Accuracy of 96.02%
[11]	EDM model	Accuracy of 90%
[12]	Radial Basis Function	ROC of 97.41%

tiveness in detecting anomalies in cloud networks, even when when there is an imbalance in class attributes.

Sohaila et al. propose a comparative analysis on various methods that have been suggested to enhance network security against malicious activities, with a focus on Intrusion Detection Systems (IDS) [11]. They use the Average One Dependence Estimators (AODE) model, Collaborative anomaly detection framework (CADF) model, Euclidean Distance Map (EDM) model, Triangle Area Based Nearest Neighbors (TANN) model, and the Naïve Bayes classifier (NB) and compare their effectiveness in finding anomalies on the 'UNSW NB15' dataset. It was found that for multi-class classification, the Online AODE model obtains an accuracy of 83.47%, the CADF model obtains an accuracy of 88%, the EDM model obtains an accuracy of 90%, the TANN model obtains 90% accuracy and the NB model receives 69.6% accuracy. Among these, the EDM model and TANN model outperforms other models.

Maria et al. focus on network traffic anomaly detection using an appropriate machine learning model, highlighting the need for safeguarding a network from malicious packets [12]. The proposed method uses machine learning models which include k-Nearest Neighbors classifier (k-NN), Fuzzy C-Means clustering (FCM), Support Vector Machine (SVM), Naïve-Bayes (NB) classifier, Radial Basis Function (RBF), and an ensemble method to identify anomalies on the 'Kyoto 2006+' dataset. Among the machine learning techniques and an ensemble method for network traffic anomaly detection on 'Kyoto 2006+' dataset, RBF classification yields the highest Rate of Change (ROC) value at 97.41%. While the Ensemble method was promising, with an ROC of 96.31%, further study in the ensemble model is encouraged.

From the above papers, which is also highlighted in Table I, it is clear that baseline machine learning models and deep learning models are not sufficient to handle network anomalies [13]. These clearly give the benefit of using a hybrid model, consisting of a combination of different baseline models used in different segments of datasets. Hence, the proposed study will be making a hybrid model with a combination of Naïve Bayes classifier and Random Forest classifier to predict anomalies more accurately giving an improvement over major baseline classifiers.

III. METHODOLOGY

A. Dataset and Preprocessing

The dataset that has been been considered for the study, is the 'UNSW-NB15_1' dataset. The initial dataset consists of 49 columns and 7000001 rows. It is reduced by one column as it had been determined not to be useful for the anomaly detection. The feature names are present in the 'NUSW-NB15_features' dataset and have been appended as the header of the chosen dataset. Some of these attributes or features are:

1) The Source IP address and port number
2) The Destination IP address and port number
3) The Transaction protocol being used
4) The state of the network
5) Record total duration
6) The service being used
7) The source jitter time in ms
8) The destination jitter time in ms
9) Recorded starting time
10) Recorded last time
11) Label output where 0 is benign and 1 is an anomaly

The state of the (-) is removed from the dataset during the cleaning process. The analysis will, hence focus on the state of the network that it is present. Now, the data has reduced to 2,52,213 rows or samples. The finalized dataset obtained, is being studied upon.

B. Naive Bayes

Naive Bayes is a very efficient algorithm in machine learning that is known for handling large datasets [14]. Its based on Bayes theorem and operates on the principle of calculating the probability of a hypothesis based on prior knowledge. The 'naive' aspect is based on the assumption that features in the dataset are considered independent of each other. Additionally Naive Bayes performs very well in various classification projects. Intially the algorithm undergoes a training phase where it learns the frequencies of features within each class from a labelled dataset. In the application phase, the learned probabilities are combined with the prior probabilities of each class to predict the most likely class for unseen data. Naive Bayes determines the category of the data by independently evaluating the probability of each feature. This method is simple and fast but unfortunately it faces difficulties especially when the features present int the training set aren't present in the testing set, which could lead to possible mis-classifications. Overall the Naive Bayes algorithm efficiency in managing large datasets makes it a favored choice in machine learning

C. Random Forest

Random forest is an ensemble machine learning method that works well with both classification and regression tasks [15].

Initially, multiple decision trees are created based on a random subset of training data which is called ”bootstrap sampling”. Additionally, for each split, the randomness is also applied to a subset of features which reduces the correlation among the trees and enhances the model's robustness. This feature selection is very helpful in giving the best accuracy [16]. Each tree grows to its maximum depth, which would most likely lead to over-fitting in individual trees but when the predictions of the trees are averaged, the over-fitting tendencies cancel out which results in a more stable prediction. For classification tasks, Random Forest algorithm employs a majority voting system and the class with the most votes across trees is chosen as the final prediction regression it averages the outputs across all the decision trees. Random Forest is known to handle complex datasets with many interrelated features, making it a feasible choice for various machine learning applications.

D. Proposed Methodology

In the proposed study a hybrid machine learning model is used for network traffic classification using the 'UNSW-NB15_1' dataset. Data preprocessing is performed initially which includes removing null values, encoding categorical variables, and standardizing numerical features. The random forest classifier is employed to select the most relevant features and after performing feature selection the Gaussian Naive Bayes model is trained on those selected features. This approach leverages the strengths of both Naive Bayes and Random Forest. The effectiveness of the models is evaluated using metrics like precision, recall, F1-score, accuracy, AUC-ROC curve and confusion matrices The methodology showcases a balanced approach to handling high-dimensional data and ensures accuracy in network traffic classification.

E. Workflow

The initial step of the analysis was to prepare the dataset. The data was cleaned; this included filling in missing values, removing duplicates and changing some columns to float type variables. Numerical and Categorical are the two types in which the data can be classified into. For the categorical data, label encoding is applied to convert non-numeric labels to numeric representations. Fig. 1 shows an attribute 'sport' which is categorical showing the range of values that it lies in. In contrast, numerical data undergoes standard scaling which normalizes it by subtracting the mean and dividing by variance. Fig. 2 shows an attribute of 'dmeanz' and the range values that lie inside the given ranges and its outliers. A feature set (X) and a target variable (y) are then made in the dataset. As a result, training sets and testing sets are divided up, so it can be used to test the model on unseen data. Now, applying Random Forest and Naïve Bayes as the classification models. To select the most impactful features, Random Forest is used first. The top 10 features are taken for further analysis as it would give the best results with the highest gini index. Naïve Bayes is more efficient when dealing with many variables and can even work with only a few important ones. Using the trained model, test results are predicted. Precision, recall,

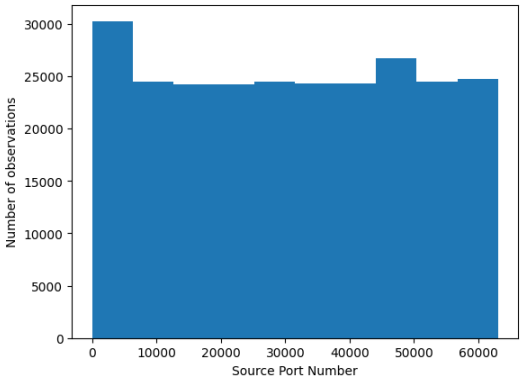

Fig. 1. The range of values for Source Port number

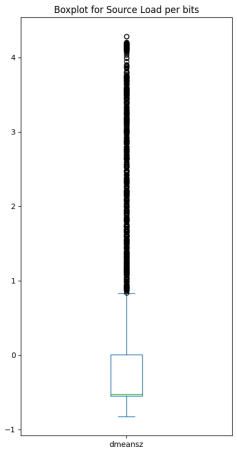

Fig. 2. The range of values for mean of packet size transmitted by the destination

F1 scores, and overall accuracy are some metrics that are used to measure effectiveness. The classification report gives a comprehensive look at how the model works by analyzing its performance across all the classes it recognizes. In addition, this gives a confusion matrix to know how well the model identifies each class. Fig. 3 shows the architecture flow of the entire system.

Fig. 3. Architecture of the Proposed Method

IV. Results

The proposed method is compared with baseline models; such as Support Vector Machine (SVM), Random Forest, and k-Nearest Neighbors (k-NN) classifier. The 'UNSW-NB15_1' dataset shows how good the model is at identifying network anomalies. With Class 0, it finds all normal network traffic instances without assuming normal activities are unusual. In addition to having a remarkable recall of 99%, this class shows it's able to distinguish between normal traffic and anomalous

traffic. Combining accuracy and precision, it gets a perfect F1-score, which shows how well it differentiates normal from abnormal.

It has a 100% recall rate when it comes to anomalous traffic (Class 1), so it identifies all anomalies in the data set correctly. In network security, it is important because one anomaly can lead to a major problem. It's good at detecting anomalies; it also gets a lot of false positives. The anomolous packets (Class 1) obtains an F1-score of 82%, so it's good at both precision and recall, leaning more towards recall. The confusion matrix in Fig. 4 shows how well the model predicts classes 0 and 1 by the proposed model. The confusion matrix in Fig. 5 shows how well the model predicts classes 0 and 1 by the Support Vector Machine (SVM) model. The confusion matrix in Fig. 6 shows how well the model predicts classes 0 and 1 by the random forest classifier model. The confusion matrix in Fig. 7 shows how well the model predicts classes 0 and 1 by the k-NN classifier model.

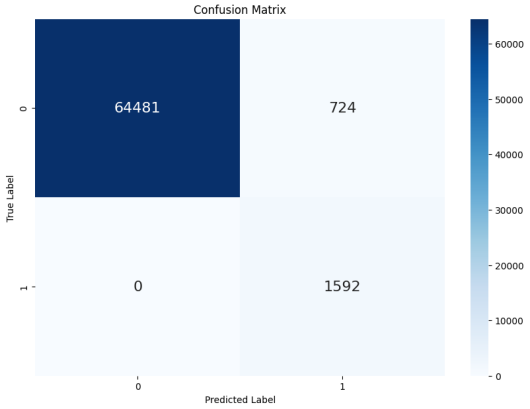

Fig. 4. The confusion matrix for the proposed hybrid model

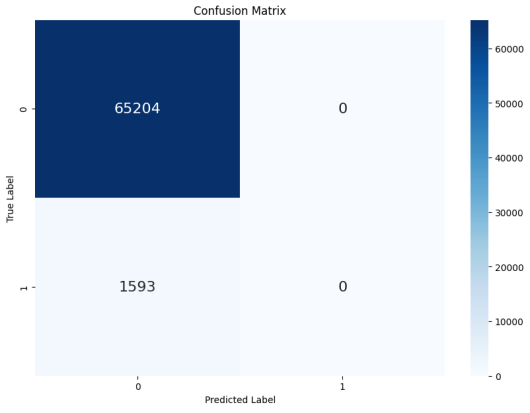

Fig. 5. The confusion matrix for the Support Vector Machine model

Fig. 8 shows the rate of convergence of the proposed method. Despite achieving a 99.03% accuracy rate, the model has a high degree of precision. Macro and weighted averages for precision, recall, and F1 score across normal and anomalous classifications show the model is good. As contrasted

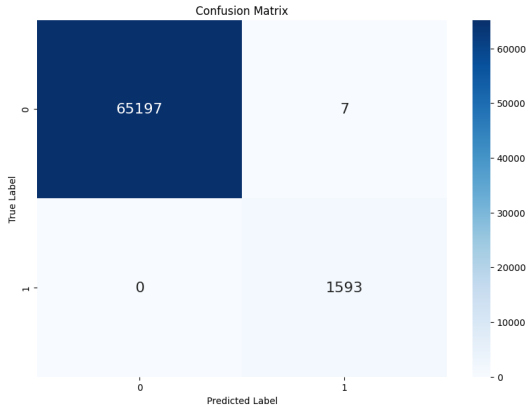

Fig. 6. The confusion matrix for the Random Forest Classifier model

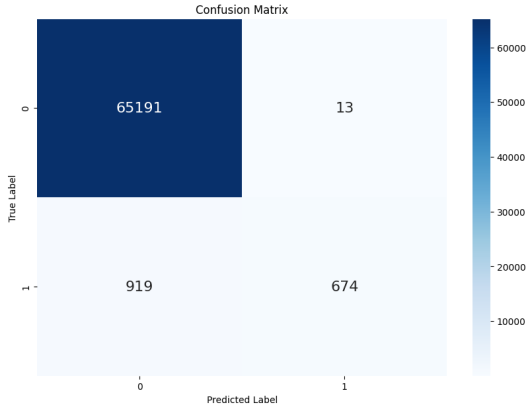

Fig. 7. The confusion matrix for the k-NN classifier model

with the large number of normal instances versus anomalies, the confusion matrix shows a lot of true positives and true negatives.

The model is capable of identifying normal network activities (Class 0) with an accuracy of 100%, and thus differentiates between typical and abnormal network behaviors without mistaking normal activities for anomalous. The recall rate of the benign packets is 99%, further shows that almost all the real cases are identified correctly by the model as being normal.

When it comes to detecting anomalies (Class 1), the model is highly sensitive with a recall rate of 100%. It ensures that every single instance, which is not part of the normal traffic, is flagged as anomalous; hence making it a key element in maintaining network security. However, there is only precision limitation on detecting anomalies by the model that stands at 69%. It implies that there tends to be many false positives which do not miss any actual anomalies but can still result in unnecessary investigations or resource allocation.

The overall accuracy of the model is relatively high; it stands at around 99.03% while F1 scores for both classes show balanced performance. Additionally, confusion matrix emphasizes on how well the model handles massive data since

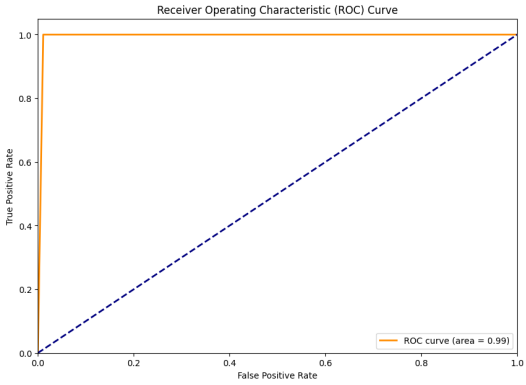

Fig. 8. Rate of Convergence for the proposed method

there exists a large number of true positive and negatives. Table II shows the scores that are obtained by all the methods, including the proposed method.

TABLE II
COMPARISON BETWEEN THE METHODS

Model Used	Accuracy	Precision	Recall	F1 Score	Time Taken
Proposed Method	99.032%	99.334%	99.032%	99.119%	9.94 s
SVM Method	97.856%	95.758%	97.856%	97.796%	298.07 s
Random Forest Method	99.989%	99.989%	99.989%	99.989%	10.65 s
k-NN Classifier	98.760%	98.756%	98.760%	98.520%	71.94 s

V. CONCLUSION

The security model excels in identifying actual anomalies, with no false negatives, meaning it does not miss any real threats. However, it does have a tendency to produce false positives, incorrectly flagging normal events as anomalies. While its accuracy in detecting genuine threats is notable, reducing these false positives is vital for improvement, especially in practical situations where correctly differentiating between true threats (anomalies) and non-threatening events is crucial.

The essence of the matter is that, although the model has shown some promising capabilities in network anomaly detection, its focus on minimizing false positives will improve its usefulness and reliability when used in real scenarios. According to performance of a model on a data set having predominantly normal traffic; it indicates its potential efficacy in similar network environments, but it needs to be taken into account whether it can suit different or more balanced datasets and in a dynamic setup.

REFERENCES

[1] K. Kavikuil and Amudha J., "Leveraging deep learning for anomaly detection in video surveillance", Advances in Intelligent Systems and Computing, vol. 815. Springer Verlag, pp. 239-247, 2019.

[2] B. Bhanu Prakash, Kaki Yeswanth, M. Sai Srinivas, Balaji S., Y. Chandra Sekhar, and Aswathy K. Nair, "An Integrated Approach to Network Intrusion Detection and Prevention", in Inventive Communication and Computational Technologies, Singapore, 2020.

[3] BP, V.K., Kusuma, S.M. and Pallavi, L.V., 2023, April. Deep machine learning based Usage Pattern and Application classifier in Network Traffic for Anomaly Detection. In 2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS) (pp. 50-54). IEEE.

[4] Satheesh, N., Rathnamma, M.V., Rajeshkumar, G., Sagar, P.V., Dadheech, P., Dogiwal, S.R., Velayutham, P. and Sengan, S., 2020. Flow-based anomaly intrusion detection using machine learning model with software defined networking for OpenFlow network. Microprocessors and Microsystems, 79, p.103285. Elsevier.

[5] Liu, Z., Thapa, N., Shaver, A., Roy, K., Yuan, X. and Khorsandroo, S., 2020, August. Anomaly detection on iot network intrusion using machine learning. In 2020 International conference on artificial intelligence, big data, computing and data communication systems (icABCD) (pp. 1-5). IEEE.

[6] Jiang, Y., Wang, W. and Zhao, C., 2019, November. A machine vision-based realtime anomaly detection method for industrial products using deep learning. In 2019 Chinese Automation Congress (CAC) (pp. 4842-4847). IEEE.

[7] Jiang, Y., Wang, W. and Zhao, C., 2019, November. A machine vision-based realtime anomaly detection method for industrial products using deep learning. In 2019 Chinese Automation Congress (CAC) (pp. 4842-4847). IEEE.

[8] Fosić, I., Žagar, D., Grgić, K. and Križanović, V., 2023. Anomaly detection in NetFlow network traffic using supervised machine learning algorithms. Journal of Industrial Information Integration, p.100466. Elsevier.

[9] Singh, R., Srivastava, N. and Kumar, A., 2021, November. Machine Learning Techniques for Anomaly Detection in Network Traffic. In 2021 Sixth International Conference on Image Information Processing (ICIIP) (Vol. 6, pp. 261-266). IEEE.

[10] Yasarathna, T.L. and Munasinghe, L., 2020, September. Anomaly detection in cloud network data. In 2020 International Research Conference on Smart Computing and Systems Engineering (SCSE) (pp. 62-67). IEEE.

[11] Eltanbouly, S., Bashendy, M., AlNaimi, N., Chkirbene, Z. and Erbad, A., 2020, February. Machine learning techniques for network anomaly detection: A survey. In 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT) (pp. 156-162). IEEE.

[12] Zaman, M. and Lung, C.H., 2018, April. Evaluation of machine learning techniques for network intrusion detection. In NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium (pp. 1-5). IEEE.

[13] P.Manjula, S.Baghavathi Priya, " An effective network intrusion detection and classification system forsecuring WS Nusing VGG-19 and hybrid deep neural network techniques", Journal of Intelligent and Fuzzy Systems,vol.43,no.5,September 2022,Scopus/SCI,10.3233/JIFS-220444,ImpactFactor:1.737

[14] A. Prakash, Anand R, S. S. Abinayaa, N. S. KalyanChakravarthy-2021 Emerging Trends in Industry 4.0 (ETI 4.0), Title of the paper: "Normalized Naïve Bayes Model to predict Type –2 Diabetes Mellitus "– 2021

[15] Thulasi Bikku,K. P. N. V. Satyasree "A Boosted Random Forest Algorithm for Automated Bug Classification ", Lecture Notes in Networks and Systems book series (LNNS,volume 650)

[16] Shanmuga Priya S. and Abinaya, M., "Feature selection using random forest technique for the prediction of pest attack in cotton crops.", International Journal of Pure and Applied Mathematics, vol. 118, pp. 2899-2902, 2018.