[1]Varinder Kaur

[2]Amandeep Kaur Virk

Hybrid Optimization Algorithm with Random Forest for Class Balancing in Network Traffic Classification

Journal of Electrical Systems

Abstract: - This study delves into the intricate task of network traffic classification employing machine learning algorithms, leveraging the KDD dataset. Initially, a range of algorithms, including KNN, Random Forest, SVM, and logistic regression, were explored, revealing Random Forest as the frontrunner in performance, albeit with its challenges. Chief among these challenges was the prevalent issue of class imbalance within the dataset. To address this critical concern, a diverse array of optimization algorithms, such as Gray Wolf, BAT, and Firefly, were meticulously examined. Moreover, a novel hybrid optimization algorithm, PSO+Gray Wolf, was developed to tackle the class imbalance intricacies inherent in the KDD dataset. The integration of this hybrid approach with Random Forest for classification yielded notably promising outcomes. The proposed model is implemented in Python and results are analyzed in terms of accuracy, precision, and recall.

Keywords: Class Imbalance, Optimization Algorithms, Machine Learning, KDD Dataset

I. INTRODUCTION

In recent years, there has been significant research interest in Network Traffic Classification (NTC) due to its critical role in cyber security and network management. Traditional methods such as port-based and deep packet inspection are rendered ineffective in encrypted communication scenarios. Consequently, machine learning (ML) technology has emerged as the most popular and efficient approach [1][2]. The academic community has witnessed the emergence of numerous well-researched studies focused on mining valuable traffic features and exploring optimal classification networks, many of which have yielded positive results. However, most investigations have overlooked two significant concerns. Firstly, the distribution of Internet traffic naturally exhibits an imbalance, with different protocols and applications generating varying proportions of traffic. Secondly, most machine learning algorithms are designed to optimize overall accuracy rather than addressing class imbalances. Consequently, classifiers tend to be biased towards the majority class, defined by the larger sample size, neglecting the minority class with fewer samples. This imbalance significantly compromises the performance of current ML-based NTC schemes when confronted with real-world imbalanced traffic classification problems [3][4].

In certain critical scenarios like intrusion detection and network censorship, where high-value traffic constitutes a small fraction of overall flow, the deterioration in performance on the minority class can be extremely detrimental. Hence, addressing the imbalance issue in NTC research becomes imperative. It is crucial to allocate sufficient focus to tackling the problem of class imbalance in network traffic classification research, especially considering its potential ramifications in contexts where high-value traffic is sparse. Class imbalance in data can manifest in two primary forms: internal and external imbalance. Internal imbalance occurs when the frequencies of classes within the data are not uniform, such as in clinical diagnosis where the majority of cases are healthy patients. External imbalance, on the other hand, stems from external influences like accumulation or storage processes [5][6]. To effectively address class imbalance, it's crucial to consider the representation of both majority and minority classes when learning from uneven data. Ensuring sufficient representation of both groups, even if they originate from non-overlapping distributions, can lead to high-quality outcomes. Strategies to address imbalance in NTC can be categorized into three primary levels: data level, algorithm level, and cost-sensitive level. At the data level, techniques involve correcting class imbalance by either under-sampling or oversampling the majority classes, or by increasing the sample size of the minority classes. Algorithm level methods utilize

[1] *Corresponding author: Department of Computer Science, Sri Guru Granth Sahib World University, Fatehgarh Sahib, Punjab varindersggswu@gmail.com

[2] Author: Department of Computer Science, Sri Guru Granth Sahib World University, Fatehgarh Sahib, Punjab Amandeep_virk@sggswu.edu.in

algorithms that reward the minority classes and penalize the majority during training. Cost-sensitive level approaches consider different misclassification costs for different classes. Among these methods, oversampling is the most frequently employed due to its clear theoretical foundation and positive results in classifying unbalanced traffic [7][8]. To address imbalance in the original dataset, oversampling involves increasing the number of samples in the minority class. Generic oversampling techniques like Synthetic Minority Oversampling Technique (SMOTE) and Random Oversampling Technique (ROS) are commonly employed to rebalance traffic data. One key advantage of these methods is their independence from the underlying classifier. However, they also come with drawbacks, such as potentially altering the original knowledge system to accommodate class populations, which could lead to the loss of crucial information or the addition of unnecessary data.

Research studies have shown that evolutionary-based methods generally outperform non-evolutionary ones in instance reduction and the analysis of unbalanced datasets [9][10]. Over time, there has been a growing interest among researchers in determining optimal solution variable values to meet specific requirements when tackling global optimization problems. Classical optimization techniques often require extensive processing resources and are prone to failure as the search space expands. However, with the introduction of meta-heuristic algorithms, such as evolutionary approaches, it becomes more computationally efficient to avoid local minima. Meta-heuristic algorithms have demonstrated their superiority in addressing challenging problems across various disciplines for several reasons. Firstly, they can circumvent local minima. Secondly, they do not require knowledge of the objective function's gradient [11][12]. Thirdly, they are simple and straightforward to construct. Lastly, they can be applied to solve a wide range of issues in diverse domains. The advancement of these algorithms is greatly facilitated by the increasing processing capacity of computers. Metaheuristic strategies draw inspiration from various metaphors such as plants, humans, birds, ecosystems, water, electromagnetic forces, and gravitational forces. Based on their behavior, metaheuristic algorithms can be categorized into four primary types: human-based, swarm intelligence-based, physics-based, and evolutionary-based.

Evolutionary-based algorithms draw inspiration from nature and typically begin by generating their population solutions randomly. One of the most famous examples is the genetic algorithm (GA) [13][15], pioneered by John Holland in the 1960s, which is founded on Darwin's theory of evolution. GA has garnered significant attention from researchers and has been refined and enhanced in various ways. It has found applications in addressing a wide range of practical problems. Additionally, this group has produced several other notable algorithms, including the memetic algorithm, genetic programming, tabu search, evolution strategy, differential evolution, flower pollination algorithm, among others. Swarm intelligence-based systems draw inspiration from the social interactions and behaviors of various organisms such as fish, birds, animals, and insects. Over the past two decades, a plethora of metaheuristic algorithms have been developed in this category, with ongoing development of more. Some researchers have devised variations of established algorithms, while others have amalgamated algorithms within this category. Notably, the particle swarm optimization (PSO) algorithm has gained prominence in this field. It takes inspiration from a flock of birds searching for the optimal spot by navigating through their search space. This algorithm was developed in 1995 by Kenneth and Eberhart, attracting significant interest due to its robust mathematical foundation for problem-solving. Other algorithms in this category include the krill herd algorithm, cuckoo search, and gray wolf optimizer. These algorithms are inspired by the rules of physics that govern the world [15][16]. Some commonly used algorithms within this category include ray optimization, gravitational search algorithms, galaxy-based search algorithms, equilibrium optimizers, and atom search optimizers.

Algorithms in the human-based category are inspired by human behaviors and activities, leveraging various human activities that affect performance to develop algorithms. Two prominent algorithms in this domain are the league championship algorithms (LCA) and teaching-learning-based optimization (TLBO).

II. Literature Review

H. Xu, et.al (2023) suggested a jumping spider optimization algorithm (JSOA) in which the Harris hawk optimization (HHO) was integrated with small hole imaging (SOI) and this method was called HHJSOA to classify network traffic [17]. Initially, the HHO escape energy factor and the hard siege strategy were employed for generating this method. This location update method was employed for improving the search range of its optimal solution and SHI for updating inferior individual. Subsequently, this method was aimed to code feature

selection problem for suggesting a JS individual coding method. Multiple iterations were helped in exploring the optimal individual employed as the selected feature for KNN algorithm. Eventually, the UNSW-NB15 and KDD99 datasets were exploited for simulating the suggested method. The experimental outcomes depicted that the suggested algorithm enhanced the accuracy up to 70.5%, fitness value around 0.00147 and number of features up to 1 on initial dataset. Additionally, this method was more convergent, superior merit-seeker, and robust while classifying network traffic.

F. Rustam, et.al (2023) introduced a comprehensive framework in which potential of Synthetic Data Augmentation Technique (S-DATE) and Particle Swarm Optimizer (PSO)-based Diverse-Self Ensemble Model (D-SEM) were utilized [18]. The M-En dataset was generated on the basis of integrating 3 dissimilar datasets such as in SDN, UNSWNB-15, and IoTID-20 for illustrating the real-time scenarios. The first technique was assisted in tackling imbalanced data distribution in given dataset, and offered higher convergence and enhanced the accuracy to detect normal and abnormal traffic. Moreover, the latter ensemble model considered the diversity taken from PSO for handling the complexity of M-En networks. In this model, the individual models, whose training was done on a subset of the M-En dataset, were helped to enhance performance. The experimental results demonstrated that the introduced framework yielded an accuracy up to 98.9%. On a statistical T-test, the second model was performed well.

A. M. Eldhai, et.al (2024) focused on enhancing performance to classify network traffic on the basis of stream learning (SL) method for selecting relevant features which mitigated load from the SDN control plane [19]. First of all, Boruta method of selecting features was presented. After that, three streaming-based approaches, known as Hoeffding adaptive trees (HAT), adaptive random forest (ARF), and k-nearest neighbor with adaptive sliding window detector (KNN-ADWIN) were deployed. These approaches were capable of handling concept drift in a dynamic way and tackling an issue related to memory and time consumption at lower overhead of SDN controller. In the end, the presented method was computed on real and synthetic traffic traces. The experimental outcomes confirmed that the presented method had offered an average accuracy of 95% and average per application concerning precision, recall, and f-score, up to 87% as compared to traditional methods. Moreover, the utilized approaches exhibited that the presented method offered an accuracy of 85%, kappa of 78%, and precision, recall, and f-score up to 62-88%. Moreover, the first approach had least time usage of 15s and memory utilization of 105KB as compared to others.

W. Liu, et.al (2022) recommended a multiclass imbalanced and concept drift network traffic classification framework based on online active learning (MicFoal) to classify network traffic [20]. In this, a configurable supervised learner was employed which initialized the network traffic classification (NTC) framework, an active learning (AL) technique with a hybrid label request method, a label sliding window group, a sample training weight (STW) formula and an adaptive adjustment (AA) mechanism for the label cost budget depending upon computing a periodic efficacy. Moreover, a new uncertain label request method relied on a variable least confidence threshold vector (VLCTV) was generated for tackling the issues of a variable multiclass imbalance ratio and even the number of classes which varied according to over time. Around 8 real-time datasets were executed for evaluating the recommended framework. The experiments proved the supremacy of recommended framework concerning efficacy as compared to existing methods while classifying traffic.

Y. Gu, et.al (2023) developed a new multi-module intrusion detection system (MIDS) called DWGF-IDS in which 3 modules were comprised to extract features, process imbalance data and detect traffic anomaly [21]. At first, a deep denoising autoencoder (DDAE) was exploited for extracting the deep feature illustration of the data and enhancing the generalized performance when the noise was inserted into AE. At second, a Wasserstein Generative Adversarial Network - Gradient Penalty (GAN-GP) optimized according to self-attention mechanism was adopted for creating few classes in the anomalous traffic. At last, this system aimed to transmit the weights and bias values of exploited model to deep neural network (DNN), which was enhanced on the basis of focal loss and employed for detecting multi-classification issue on least dimensional balanced traffic data. The NSL-KDD and CSE-CIC-IDS-2018 datasets were employed for computing the developed system. The developed system offered an accuracy of 85.05% on first dataset and 99.57% on second dataset for mutli-classification. The experimental results indicated that the developed system was robust for mitigating higher dimensionality and imbalance of IoT data, enhancing detection rate (DR) of unknown assaults, and improving misclassification of rare classes of attack traffic.

J. Koumar, et.al (2024) projected a new expanded IP flow known as Network Time Series Analysed (NetTiSA) flow when the time series of packet sizes was analyzed [22]. Twenty-five diverse tasks were tested to compute whether this method was applicable and practical. This method was implemented practically on the basis of sizes of flows which its features had expanded and computing the efficacy impacts of their computational in the flow exporter. The new features were proved less expensive and offered efficacy concerning discriminability. The analysis indicated the supremacy of trained machine learning (ML) over other methods. Lastly, the projected method was capable of bridging the gap and offered features which were cost-effective, and small in size to classify traffic. It scaling was done to wide monitoring structures, that offered ML traffic classification even to 100 Gbps backbone lines. The results confirmed that the projected method was effective.

X. Yan, et.al (2024) established a High-speed Encrypted Traffic Classification (HETC) technique in which 2 phases to classify traffic [23]. Firstly, the encryption of traffic was detected utilizing randomly sampled short flows and extracting aggregation entropies with chi-square test features so that diverse patterns of the byte composition and distribution were computed amid encrypted and unencrypted flows. Secondly, the binary features were presented relied on earlier features and classifying traffic when these payload features were classified a Random Forest (RF) algorithm. The experiment outcomes depicted that the established technique had offered F-measure up to 94% to detect encrypted flows and a 85%–93% to classify fine-grained flows on 1-KB flow-length dataset as compared to other methods. Moreover, this technique had no longer waiting time for the end of the flow and applicable for extracting mass computing features. Its average time for processing every flow was only 2 or 16 ms, that was lower in contrast to the flow duration in most cases. Hence, this technique was proved robust to classify traffic at higher speed.

F. Ullah, et.al (2023) investigated an Intrusion Detection System using transformer-based transfer learning for Imbalanced Network Traffic (IDS-INT) [24]. The transformer-based transfer learning (TTL) method was helped to learn feature interactions in representing network feature and imbalanced data. This system was employed to collect comprehensive information of every kind of attack from network interaction descriptions having network nodes, assault kind, reference, host information, etc. Later, TTL was developed to learn the detailed feature illustration in accordance with their semantic anchors. Moreover, this system aimed to balance the abnormal traffic and detect minority attacks. For this, Synthetic Minority Oversampling Technique (SMOTE) was implemented. The Convolution Neural Network (CNN) algorithm was utilized to extract deep attributes from the balanced network traffic. At last, the hybrid of Convolution Neural Network-Long Short-Term Memory (CNN-LSTM) had detected various kinds of assaults from the deep features. The experiments on UNSW-NB15, CIC-IDS2017, and NSL-KDD datasets depicted that the the investigated system was worked effectively to classify traffic.

X. Wang, et.al (2021) suggested a neural architecture search (NAS) on the basis of multi objective evolutionary algorithms (MOEAs) which classified malicious network traffic [25]. This method was implemented for simplifying search space when the spatial ratio and number of channels of the model were mitigated. Moreover, a variation was found in search strategy in the effectual search space, and the EAs were implemented with the nondominated sorting genetic algorithm with the elite retention strategy (NSGA-II), strength Pareto evolutionary algorithm (SPEA-II) and multi-objective particle swarm optimization (MOPSO) for addressing the developed MO-NAS. An analysis was conducted on population convergence times, accuracy, Pareto optimality sets (POSs), complexities and running speeds of these method and the NSGA-II based method was proved more robust. The experimental results depicted that the suggested approach was performed more effectively on 2 public datasets at least computation complexity concerning FLOPs. The F1score of this approach was found 99.806% and 99.369% with 11.501 MB on IDS2012 and 4.718 MB FLOPs on ISCX VPN dataset.

J. Qin, et.al (2022) introduced a novel imbalanced encrypted traffic classification (IETC) algorithm to classify traffic [26]. This algorithm was planned on the basis of enhanced convolutional block attention module (CBAM) and re-weighted cross-entropy focal loss (CEFL) function. The redefined imbalance degree was utilized for creating a weight function, so that the weights of the categories were re-assigned. The initial one was relied on the redefined imbalance degree (RID) for considering the attributes of the minority samples, and enhancing the illustration power of these samples. The latter one was assisted in expanding the loss gap amid minority and majority samples. The ISCX Tor 2016 dataset was executed for computing the introduced algorithm. The

experiments demonstrated that the introduced algorithm was more effective in comparison with other method and led to enhance the precision of the minority categories up to 93.28%, recall of 91.71%, and F1 score of 92.49%.

III. RESEARCH METHODOLY

The network traffic classification models will help us to identify type of traffic in the network. The network traffic classification models have various steps which include data set pre-processing, feature extraction, classification and performance analysis. The various schemes are proposed in the past years for the efficient network traffic classification. The existing schemes has various drawbacks which we need to entertain in the research work. The KDD dataset is very large in size and has the problem of class unbalancing due to which existing schemes are unable to achieve desired performance. In this research work, hybrid optimization algorithm is proposed which solve class unbalancing problem and improve model performance. The motivation of this research work is to increase accuracy and methodology is described below: -

A. Dataset Input

The initial stage is the dataset input in which data gathered from the genuine source named KDD is utilized for input. This study employs a NSL-KDD dataset in which 42 attributes are compromised. The duplicate instances are eliminated to enhance the KDD'99 datasets with the purpose of removing the biased classification results from the dataset. The utilization of only 20% of training data is done. However, various editions of the data set are present. This data is represented in the form of KDDTrain+_20Percent.

B. Data Pre-processing

In this phase of research work, the KDD dataset has the problem of class unbalancing. The hybrid optimization algorithm is proposed which is the combination of gray wolf algorithm and PSO algorithm to balance KDD dataset. The Grey Wolf Optimization algorithm is a novel swarm intelligence algorithm planned on the basis of grey wolf management hierarchy and group hunting nature. The notion of this algorithm is very simple, its parameters are few, programming is executed easily, and distributed parallel computing and strong global search potentials are supported in this algorithm. Thus, it becomes extensive in global optimization problems under the domain of computer science (CS), engineering science, and management science. Naturally, grey wolves are eager to animate in containers and follow a strict social hierarchy. A pack of consisted of 4 kinds of wolves, and ranks are assigned to them from uppermost to lowermost in the social hierarchy: the α wolf, β wolf, δ wolf, and ω wolf. Moreover, this algorithm is also depending upon the social hierarchy of grey wolves and their preying behavior, and its specific mathematical model is defined as:

1) Surround the prey: The procedure of hunting initially aimed at surrounding the prey. The major objective is of computing the distance amid current grey wolf and the prey, and updating the position in accordance with distance. The behaviour of grey wolves to surround the target is expressed as:

$$X(t+1) = X_P(t) - A \times D \quad (1)$$

And

$$D = |C \times X_P(t) - X(t)| \quad (2)$$

In this, equation (1) denotes the updating process of location of grey wolf, and equation (2) illustrates a formula to compute the distance amid grey wolf individual and prey. The current iteration number is denoted with t, $X_P(t)$ is used to denote the current position vectors of the prey and $X(t)$ for grey wolf at iteration t. A and C are employed for denoting coefficient vectors which are computed via equations (3) and (4) as:

$$A = 2 \times a \times r_1 - a, \quad (3)$$

$$C = 2 \times r_2, \quad (4)$$

And

$$a = 2 - 2 \times \frac{t}{t_{max}} \quad (5)$$

In this, a denotes the convergence factor, and a linear alleviation is found from 2 to 0 after maximizing the number of iterations. r_1 and r_2 are employed to show random vectors in [0, 1]. The equation (5) is used to define the computation formula a and t_{max} is used to define the extreme number of iterations.

2) Hunting: An abstract search space has not any specific place of optimal solution. For simulating the hunting behavior of grey wolves, 3 wolves: α, β, and δ are considered which may have a superior understanding of the potential location of prey. The initial one is employed as an optimal solution, latter as the sub-optimal solution, and last one as 3rd optimal solution. Other Gray wolves are capable of updating their positions on the basis of α, β, and δ wolves, and their computation expressions are defined as:

$$D_\alpha = |C_1 \times X_\alpha - X(t)|$$
$$D_\beta = |C_2 \times X_\beta - X(t)|$$
$$D_\delta = |C_3 \times X_\delta - X(t)| \quad (6)$$
$$X_1 = |X_\alpha \times A_1 - D_\alpha|$$
$$X_2 = |X_\beta \times A_2 - D_\beta|$$
$$X_3 = |X_\delta \times A_3 - D_\delta| \quad (7)$$

And

$$X(t+1) = \frac{(X_1 + X_2 + X_3)}{3} \quad (8)$$

In this, D_α is employed to illustrate the distance amid current grey wolf and α wolf; D_β for the distance from current to β wolf; D_δ is used to define the distance amid current and δ wolf; and X_α is a position vectors of α wolf, X_β is of β wolf, and X_δ is of δ wolf. $X(t)$ shows the current location of grey wolf. The random vectors are represented with C_1, C_2, and C_3 and equation (4) is executed to compute them. Equation 3 is executed to compute A_1, A_2, and A_3. In addition, the step length and direction of grey wolf individuals to α, β, and δ wolves are evaluated with equation (7), and equation (8) is used as the position-updating expression used by grey wolf individuals. The PSO algorithm is planned in accordance with the particle behavior, like flocking, swarming and herding. Every particle is capable of changing its flight based on the self or previous experience of flight of its companion. The place of food is known to it due to its own experience and this location as personal best position with (P). Moreover, the particle is considered as the finest position in case a swarm is defined as global best position (G) [5]. This algorithm aims to reconstruct this phenomenon for addressing the problems of real time. In addition, the particles are assisted in generating a swarm. These particles are useful to fly arbitrarily in the solution space at velocity v_i at position x_i and change their places relied on the personal experience, social and cognitive nature. The position and velocity of every particle i at tth generation are defined as:

$$v_i(t+1) = wv_i + c_1 r_1\big(P_i(t) - x_i(t)\big) + c_2 r_2\big(G(t) - x_i(t)\big) \quad (9)$$
$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (10)$$

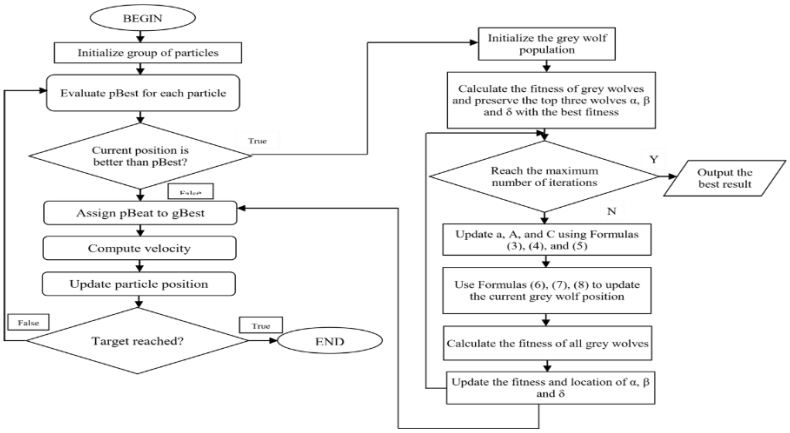

Fig. 1. Hybrid Optimization Algorithm for Class Balancing

C. Feature Selection and Classification

In the proposed model the feature selection is done manually and a scheme of random forest is applied for the classification. In Random Forest the trees are combined to create a single, strong learner after averaging or getting the majority vote when numerous tiny, weak DTs are formed in tandem. The RFs are frequently studied as the most precise learning algorithms for training. Formally, an RF is a predictor built of a set of randomly generated base regression trees, where $\{r_n(x,\ominus_m, D_n), m \geq 1\}, where\ \ominus_1,\ominus_2$ .... are the independently distributed outputs of a randomly generated variable $\ominus$. These RT integrations are performed to create the aggregated regression estimate.

$$\bar{r}_n(X, D_n) = \mathbb{E}_{\ominus}[r_n(X,\ominus, D_n)] \qquad (11)$$

In where subject to X and the data set D_n, $\mathbb{E}_{\ominus}$ denotes what is expected as a function of the random parameter. The dependence of the estimations would be eliminated from the sample in the following notation to simplify it a little and given in the form $\bar{r}_n(X)$ rather than $\bar{r}_n(X, D_n)$. When the M RTs are generated and the average of the individual outcomes is obtained, Monte Carlo is used to calculate the expectation above. When creating individual trees, where the choice of the split coordinate and split position are constructed, the randomizing variable $\ominus$ is used to assess how well subsequent cuts work. As the independent of X and the training sample D_n, the variable $\ominus$ is inferred.

IV. RESULT & DISCUSSION

In this section includes results of the proposed model which is compared with existing models for the network traffic classification. In the Below sections dataset details with results are elaborated: -

A. Dataset Description

The KDD 99 intrusion detection datasets are based on the 1998 DARPA initiative to provide designers of intrusion detection systems (IDS) with a benchmark on which to evaluate different methodologies. To do so, a simulation is made of a factitious military network consisting of three 'target' machines running various operating systems and services. Additional three machines are then used to spoof different IP addresses, thus generating traffic between different IP addresses. Finally, there is a sniffer that records all network traffic using the TCP dump format. The total simulated period is seven weeks. Normal connections are created to profile that expected in a military network and attacks fall into one of four categories: User to Root; Remote to Local; Denial of Service; and Probe. • Denial of Service (dos): Attacker tries to prevent legitimate users from using a service. • Remote to Local (r2l): Attacker does not have an account on the victim machine, hence tries to gain access. • User to Root (u2r): Attacker has local access to the victim machine and tries to gain super user privileges. • Probe: Attacker tries to gain information about the target host. In 1999, the original TCP dump files were pre-processed for utilization in the Intrusion Detection System benchmark of the International Knowledge Discovery and Data Mining Tools Competition. To do so, packet information in the TCP dump file is summarized into connections. Specifically, "a connection is a sequence of TCP packets starting and ending at some well-defined times, between which data flows from a source IP address to a target IP address under some well-defined protocol.

B. Results

This research work is conducted on the basis of classifying the network traffic. The framework of classifying the network traffic classification is executed in diverse phases in which the data is pre-processed, features are extracted and the data is classified. The dataset which is used for the model testing is of KDD.

The KDD dataset has the 42 attributes and target set which contain multiple classes of different attacks. Various metrics such as accuracy, precision and recall are considered to evaluate the introduced technique. The Classification implemented for the network traffic classification. The SVM, KNN, Logistic Regression and Random Forest is implemented for the network traffic classification. The results of the classification algorithm is described in table 1

Table 1. Machine Learning Models Without Class balancing

Model	**Accuracy**	**Precision**	**Recall**
SVM Classifier	75.74 Percent	81 Percent	76 Percent
Logistic Regression	72.67 Percent	80 Percent	77 Percent
KNN Classifier	70 Percent	72 Percent	76 Percent
Random Forest Classifier	75.78 Percent	76.89 Percent	75.90 Percent

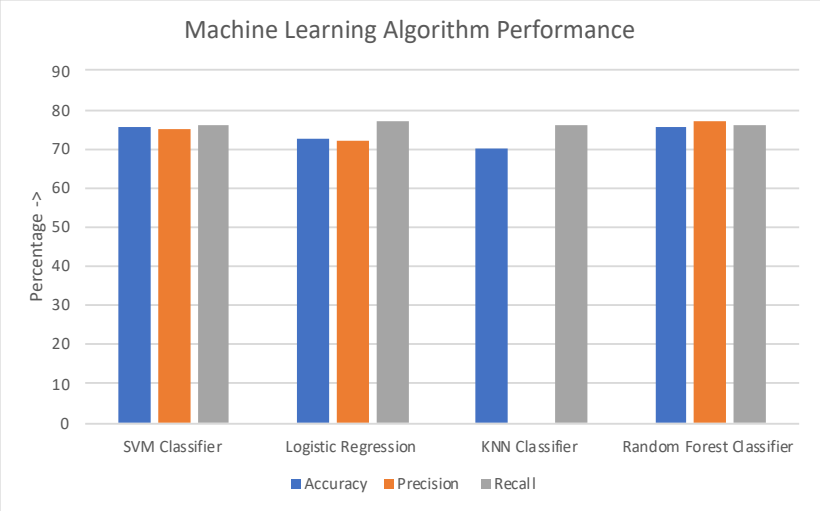

Fig. 2. Machine learning Algorithm Performance

As shown in figure 2, the performance of machine learning algorithm is illustrated when the class imbalance problem is not resolved. It is analyzed from the results that' the random forest algorithms give high accuracy, precision and recall as compared to SVM, Logistic regression and KNN for the network traffic classification.

Table 2. Random Forest Model with Class Balancing.

Models	Accuracy	Precision	Recall
Gray Wolf+ Random Forest	73.88 Percent	81 Percent	74 Percent
BAT +Random Forest	76.64 Percent	82 Percent	77 Percent
Firefly+ Random Forest	77.36 Percent	77 Percent	74 Percent
PSO+ Gray Wolf+ Random Forest	99.76 Percent	99 Percent	99 Percent

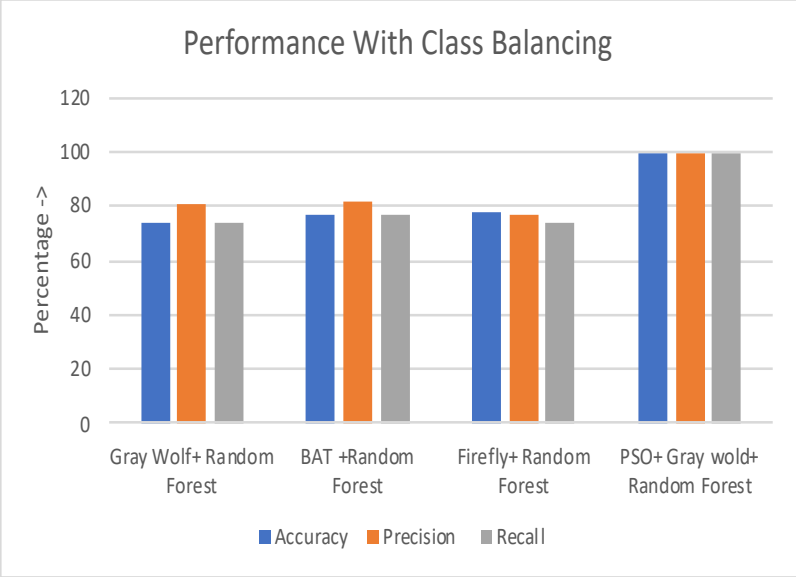

Fig. 3. Performance Analysis with Class Balancing

As shown in figure 3, It is analyzed from the previous results that's random forest is the best-performing machine learning model for network traffic prediction. The random forest is applied with different optimization algorithms like Gray Wolf, BAT, and Firefly for the class balancing. It is analyzed that hybrid optimization algorithm which is PSO+ Gray wolf when applied with random forest for the classification it gives the highest performance in terms of accuracy, precision and recall.

CONCLUSION

In conclusion, our work focused on classifying network traffic using machine learning algorithms with the KDD dataset. Initially, we found that Random Forest performed the best among KNN, SVM, and logistic regression, achieving approximately 76% accuracy. However, we identified a class imbalance issue in the dataset. To address this problem, we explored various optimization algorithms such as Gray Wolf, BAT, and Firefly. Additionally, we devised a novel hybrid optimization algorithm, PSO+Gray Wolf, which effectively tackled the class imbalance in the KDD dataset. When we applied this hybrid algorithm in conjunction with Random Forest for classification, we observed a significant improvement in accuracy, reaching approximately 99%. This notable increase in accuracy, around 30% higher than other models, demonstrates the effectiveness of our proposed approach in overcoming class imbalance challenges in network traffic classification.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil M. Sigala, A. Beer, L. Hodgson H. Shi, H. Li, D. Zhang, et al., "An efficient feature generation approach based on deep learning and feature selection techniques for traffic classification," Computer Networks, vol. 132, pp. 81-98, 2018.

[2] C. Liu, L. He, G. Xiong, et al., "Fs-net: A flow sequence network for encrypted traffic classification," in IEEE INFOCOM 2019IEEE Conference on Computer Communications. IEEE, 2019, pp. 1171-1179.

[3] S. E. Gomez, L. Hernández-Callejo, B. C. Martínez, et al., "Exploratory study on class imbalance and solutions for network traffic classification," Neurocomputing, vol. 343, pp. 100-119, 2019.

[4] M. Lotfollahi, M. J. Siavoshani, R. S. H. Zade, et al., "Deep packet: A novel approach for encrypted traffic classification using deep learning," Soft Computing, vol. 24, no. 3, pp. 1999-2012, 2020.

[5] P. Wang, S. Li, F. Ye, et al., "PacketCGAN: Exploratory Study of Class Imbalance for Encrypted Traffic Classification Using CGAN," arXiv preprint arXiv:1911.12046, 2019.

[6] R. Hasibi, M. Shokri, M. Dehghan, "Augmentation scheme for dealing with imbalanced network traffic classification using deep learning," arXiv preprint arXiv:1901.00204, 2019.

[7] Z. Liu et al., "Self-paced Ensemble for Highly Imbalanced Massive Data Classification," 2020 IEEE 36th International Conference on Data Engineering (ICDE), 2020, pp. 841-852

[8] Y. Park and J. -S. Lee, "A Learning Objective Controllable Sphere-Based Method for Balanced and Imbalanced Data Classification," in IEEE Access, vol. 9, pp. 158010-158026, 2021

[9] H. Shamsudin, U. K. Yusof, A. Jayalakshmi and M. N. Akmal Khalid, "Combining oversampling and undersampling techniques for imbalanced classification: A comparative study using credit card fraudulent transaction dataset," 2020 IEEE 16th International Conference on Control & Automation (ICCA), 2020, pp. 803-808

[10] F. Feng, K. -C. Li, J. Shen, Q. Zhou and X. Yang, "Using Cost-Sensitive Learning and Feature Selection Algorithms to Improve the Performance of Imbalanced Classification," in IEEE Access, vol. 8, pp. 69979-69996, 2020

[11] B. Krawczyk, A. Cano and M. Woźniak, "Selecting local ensembles for multi-class imbalanced data classification," 2018 International Joint Conference on Neural Networks (IJCNN), 2018, pp. 1-8

[12] S. Sridhar and A. Kalaivani, "A Two-Tier Iterative Ensemble Method To Tackle Imbalance In Multiclass Classification," 2020 International Conference on Decision Aid Sciences and Application (DASA), 2020, pp. 1248-1254

[13] A. Abdullah ALFRHAN, R. Hamad ALHUSAIN and R. Ulah Khan, "SMOTE: Class Imbalance Problem in Intrusion Detection System," 2020 International Conference on Computing and Information Technology (ICCIT-1441), 2020, pp. 1-5

[14] Q. Kang, X. Chen, S. Li and M. Zhou, "A Noise-Filtered Under-Sampling Scheme for Imbalanced Classification," in IEEE Transactions on Cybernetics, vol. 47, no. 12, pp. 4263-4274, Dec. 2017

[15] H. A. Gameng, B. B. Gerardo and R. P. Medina, "Modified Adaptive Synthetic SMOTE to Improve Classification Performance in Imbalanced Datasets," 2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS),

[16] 2019, pp. 1-5

[17] W. Fang, X. Gong, G. Liu, Y. Wu and Y. Fu, "A Balance Adjusting Approach of Extended Belief-Rule-Based System for Imbalanced Classification Problem," in IEEE Access, vol. 8, pp. 41201-41212, 2020

[18] H. Xu, Y. Hu, W. Cao and L. Han, “An Improved Jump Spider Optimization for Network Traffic Identification Feature Selection”, Computers, Materials & Continua, vol. 76, no. 3, pp. 3239-3255, 2023, doi: 10.32604/cmc.2023.039227.

[19] F. Rustam and A. D. Jurcut, “Malicious traffic detection in multi-environment networks using novel S-DATE and PSO-D-SEM approaches”, Computers & Security, vol. 136, pp. 85-94, 26 October 2023, doi: 10.1016/j.cose.2023.103564.

[20] A. M. Eldhai et al., "Improved Feature Selection and Stream Traffic Classification Based on Machine Learning in SoftwareDefined Networks," in IEEE Access, vol. 12, pp. 34141-34159, 2024, doi: 10.1109/ACCESS.2024.3370435.

[21] W. Liu, C. Zhu and Q. Liu, “Multiclass imbalanced and concept drift network traffic classification framework based on online active learning”, Engineering Applications of Artificial Intelligence, vol. 117, pp. 12-20, 24 November 2022, doi: 10.1016/j.engappai.2022.105607.

[22] Y. Gu, Y. Yang and M. Gao, “Learning-based intrusion detection for high-dimensional imbalanced traffic”, Computer Communications, vol. 212, pp. 366-376, 24 October 2023, doi: 10.1016/j.comcom.2023.10.018.

[23] J. Koumar, K. Hynek and T. Čejka, “NetTiSA: Extended IP flow with time-series features for universal bandwidth-constrained high-speed network traffic classification”, Computer Networks, vol. 12, pp.56-63, 3 January 2024, doi: 10.1016/j.comnet.2023.110147.

[24] X. Yan, L. He and G. Xie, “High-speed encrypted traffic classification by using payload features”, Digital Communications and Networks, vol. 2, pp.85-92, 28 February 2024, doi: 10.1016/j.dcan.2024.02.003.

[25] F. Ullah, S. Ullah and J. Chun-Wei Lin, “IDS-INT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic”, Digital Communications and Networks, vol. 1, no. 12, pp. 287-295, 17 March 2023, doi: 10.1016/j.dcan.2023.03.008.

[26] X. Wang et al., "Evolutionary Algorithm-Based and Network Architecture Search-Enabled Multiobjective Traffic Classification," in IEEE Access, vol. 9, pp. 52310-52325, 2021, doi: 10.1109/ACCESS.2021.3068267.

[27] J. Qin, G. Liu and K. Duan, "A New Imbalanced Encrypted Traffic Classification Model Based on CBAM and Re-Weighted Loss Function", Applied Sciences, vol. 12, pp. 9631-9638, 2022, doi: 10.3390/app12199631.

BIOGRAPHY OF AUTHORS

Ms. Varinder Kaur is a dedicated Research Scholar and Assistant Professor at Sri Guru Granth Sahib World University, Fatehgarh Sahib, Punjab. Currently pursuing her Ph.D. in Computer Science, she specializes in Network Traffic Classification using advanced Machine Learning techniques. Ms. Kaur earned her Master's in Computer Applications (2018) from the same institution, following her Bachelor's in Computer Applications from Kurukshetra University. Since 2021, she has been passionately contributing to academia, guiding and inspiring students in Computer Science. For inquiries, she can be reached at varindersggswu@gmail.com.

Contact No: +91 9170000025

Dr. Amandeep Kaur Virk is working as an Assistant Professor at Sri Guru Granth Sahib World University, Punjab, and has teaching experience of more than 13 years. She has done her M.Tech in Computer science from NIT, Jalandhar, Punjab, and her Ph.D. from Punjabi University, Patiala, Punjab. Her areas of specialization include computer networks, algorithms, metaheuristics, and machine learning. She can be contacted at email: amandeep_virk@sggswu.edu.in