[1]V S Stency

[2]Dr. N. Mohanasundaram

[3]Dr. R. Santhosh

# A Hybrid Light Gradient Boosting Approach with Deep Boltzmann Machine for Intrusion Detection System

***Abstract:*** - The need for network attack analysis and the number of cyber threats and attacks are increasing. Due to the scalability and adaptability of its internet-based computing resources, cloud computing is gaining popularity among businesses worldwide. Scientists are increasingly interested in cloud data security and face the difficulty of protecting hosts, enterprises, and data against more advanced digital threats. Over the past few decades, researchers have experimented with the Intrusion Detection (ID) paradigm, leading to various methodologies. However, critical to analyze the abnormalities in the intrusion detection framework. This research aims to classify the attacks or abnormalities in the network included in the NSL-KDD dataset using a Hybrid Light Gradient Boosting approach with Deep Boltzmann Machine. A Deep Boltzmann Machine Classifier and an ensemble model of the Light Gradient Boosting technique were used to create this model. Gradient boosting techniques improve the performance of DL classifiers by minimizing the number of errors discovered during network intrusion detection. This work evaluates and contrasts the proposed classifier with recognized classification strategies. The proposed model surpasses previous Recall, F-Measure, Precision, and Accuracy techniques.

***Keywords:*** Light Gradient Boosting Approach, Machine Learning, Intrusion Detection, Deep Boltzmann Machine Classifier, Ensemble classifier

## 1. Introduction

Industry, business, and other human life extensively use computer networks. Many attacks affect computer networks' availability, integrity, and confidentiality [1]. Institutions and companies are continually increasing their spending on cybersecurity technology to provide their users with safe and stable service. One of the most prevalent and severe attacks is denial-of-service (DOS)[2]. DOS assaults are primarily directed at web services and social networking sites. R2L attacks, such as SPY and PHF, try to prepare illicit access to network resources. Middle-boxes like firewalls, intrusion detection systems (IDS) and antivirus are used in security solutions [3]. A firewall prohibits access between networks, thereby limiting access from one network to the next. In the event of an internal attack, it does not, however, issue any notifications. Specific defence strategies like Artificial intelligence-based IDS—must be created to ensure system security. This security tool examines the system for unusual behaviour while keeping track of network traffic and alerting the system or network administrator. In addition, IDS tools are frequently used to increase sensitivity to cyber threats. IDS are divided into two categories, such as network-based and host-based [4]. Network and information security are the main issues facing the expanding economy. The Intrusion Detection System (IDS) would alert the administrator if it discovered any intrusions. In this context, two possible attack types are signature-based attacks and profile-based assaults. A method of data analysis Machines Learning [5] performs automated analytical model creation. According to one of the artificial intelligence concepts, a machine can be trained, make decisions, and learn to recognize patterns with minimal human involvement.

Supervised and unsupervised learning are the two methods of ML that are most frequently utilized. To reduce the analyst's workload, machine learning algorithms also suggest minimizing false positives, which take up processing time [6]. In contrast, implementing ML techniques in the context of network ID has resulted in the introduction

[1] Research Scholar, Department of CSE, Karpagam Academy of Higher Education,

Coimbatore - 641021, Tamil Nadu, India.

2Professor, Department of CSE, Faculty of Engineering, Karpagam Academy of Higher

Education, Coimbatore - 641021, Tamil Nadu, India.

3Professor, Department of CSE, Faculty of Engineering, Karpagam Academy of Higher

Education, Coimbatore - 641021, Tamil Nadu, India.

E-mail: stenz.denz@gmail.com

Copyright © JES 2024 on-line : journal.esrgroups.org

of hostile machine learning, which poses a fundamental security concern. Several popular machine learning (ML) techniques for ID include Nave Bayes, the support vector machine (SVM), K-nearest neighbour (KNN), and neural networks. Classifier ensembles and single classifiers like CART and MLP [7] are evaluated using metrics and area under the receiver operating characteristic curve (AUC). IDS-ML offer a learning-based method for discovering attack classes based on learnt normal and attack behaviour. ML-based intrusion detection systems (based on supervised learning techniques) aim to develop a general representation of recognized threats. Systems for detecting misuse are unable to identify unidentified attacks. These tactics have a high detection accuracy for well-known attacks, but regularly maintaining the signature database required by this intrusion detection system contributes to the User's workload [8].

The paper is divided into the sections listed below, The introduction is provided in Section I, and Section II highlights our unique contributions while comparing our work to related surveys. Following a review of the suggested approach in Section III, numerous machine learning techniques are discussed along with their characteristics. Results and discussion of the suggested model are described in Section IV. At last, the conclusion and suggested next steps are found in Section V, and they offer a summary of the ongoing and future research work.

## 2. Literature Review

The security method of the intrusion detection tree (IntruDTree) with ML was presented by Sarker, Iqbal H., et al. [9]. Before constructing a tree-based extended ID model with the chosen key features, they first analyze the significance ranking of security factors. By lowering the number of feature dimensions, this model increases prediction accuracy for test instances that are not observed and lowers model computing complexity. Finally, we ran tests on cyber security datasets and computed precision, f1 score, ROC, and recall values to calculate the IntruDTree method’s performance. To evaluate the usage of the outcome of the security model, contrast the outcomes of the IntruDTree model and a variety of well-known classical machine learning techniques. Future research could examine the performance of the IntruDTree model at the application level in the field of cyber security, collecting Massive datasets with new security components in IoT security services.

In Almseidin, Mohammad et al. [10] presented some experiments conducted and reviewed to evaluate different ML classifiers corresponding to the KDD intrusion dataset. Several tests were conducted and evaluated to determine the efficacy and effectiveness of the ML classifiers Random Forest (RF), Bayes Network (BN), Random Tree (RT), Naive Bayes (NB), J48, and MLP. The KDD intrusion detection dataset was utilized for all tests. There are around 79% DOS attacks, Ninteen Percentage regular packets, and two percentage other attacks inside the KDD dataset (R2l, U2R and PROBE). Study results demonstrated that no single ML system could effectively combat all cyberattacks. The decision table (rules-based classifiers) had the lowest false negative value (0.002) but not the highest detection accuracy rate. Additionally, the true positive (TP) and average accuracy rates are insufficient to identify the intrusion to safeguard the confidentiality and availability of network resources.

Alqahtani, Hamed, et al. [11] examined the efficacy of the data-driven ID model by including general classification approaches in machine learning. For security objectives, IT personnel, e-commerce, and application developers are highly concerned about the viability and usefulness of ML-based Intrusion Detection System (IDS)modelling. Typically, A collection of cyber-security data includes several cyberattack types with the necessary features. Consequently, specific classifiers could have had better accuracy and real prediction rate depending on various attack types and variables. They assessed some performance criteria, including recall, accuracy, f1-score, and precision. It aims to expand the cyber-security datasets and develop an intrusion detection system (IDS) generated by data for the community's automated security services.

Jamadar, Riyazahmed, et al. [12] proposed a methodology for constructing a network IDS utilizing the decision tree machine learning technique. However, the network and attack tactics have changed dramatically, necessitating the usage of the CICIDS 2017 Dataset instead of KDDCup99. Therefore, it can detect attacks based on the present state of the network. The Decision Tree-based technique for developing an efficient intrusion detection model is shown and analyzed. The experimental analysis indicates that the suggested method can create a high detection rate (HDR), accuracy and low False-Positive-Rate (FPR).

The AB-TRAP design, which allows enhanced network traffic and considers organizational considerations to ensure the entire adoption of the answer, is covered by Bertoli, Gustavo De Carvalho, and associates [13]. The five-step AB-TRAP methodology entails the following steps: I. creating the dataset according to attack; II. creating the real dataset; III. training ML models; IV. putting the models into practice (realizing the models), and V. assessing the effectiveness of the deployed model. These frameworks' important characteristics are reproducibility, utilization of the most recent network traffic and threats, and attention to deployment and implementation issues. Power consumption is a critical performance indicator for battery-aware systems, and disadvantages will be considered.

Alhajjar, Elie, et al. [14] examined the adversarial nature of Network Intrusion Detection Systems (NIDS). They concentrate on the attack perspective, which covers approaches for generating negative samples capable of escaping various machine learning models. They investigate the development of negative examples using deep learning (generative adversarial networks) and evolutionary computing (particle swarm optimization and genetic algorithms). They test these methods on two openly accessible data sets, the NSL-KDD and UNSW-NB15, and compare their effectiveness in avoiding a NIDS to a standard perturbation methodology, Monte Carlo simulation. The findings show that adversarial example-creation strategies lead to huge misclassification rates in eleven machine-learning models and one voting classifier. This research demonstrates the susceptibility of machine learning (ML)-based NIDS to adversarial disruption.

The NIDS of the SDN controller, Alzahrani, Abdullah O., et al. [15] suggests utilizing machine learning (ML) methods for traffic investigation to identify malicious network behaviour. Numerous tree-based ML techniques, such as DT, RF, and XG Boost, indicate attack detection. The benchmarking dataset for numerous cutting-edge NIDS algorithms, the NSL-KDD dataset, is utilized for training and calculating the suggested approaches. The dataset is subjected to several sophisticated preprocessing algorithms to extract the proper data format, producing results superior to those of other systems. The dataset is subjected to several complex preprocessing algorithms to extract the proper data format, producing results superior to those of other systems. Using only five of the 41 NSL-KDD attributes, a multi-class classification job is accomplished with a 95.95% accuracy by detecting an attack's presence and describing the attack type (DDoS, PROBE, R2L, and U2R).

Megantara, Achmad Akbar, and colleagues [16] presented a hybrid machine-learning strategy for developing a workable model by integrating the data reduction and feature selection methods, representing an unsupervised learning method. The Local Outlier Factor (LOF) approach was used to identify abnormal or outlier data. A decision tree-based method and recursive feature reduction were utilized to choose pertinent and crucial features. As a result, the suggested strategy is superior to most of the previous research in the NSL-KDD dataset in detecting R2L with the best accuracy (99.89%). It maintains superior accuracy for other attack types. Its performance is hence more reliable than the others. Binary classifications present extra difficulties for the UNSW-NB15 dataset. Additionally, this approach may shorten processing time.

Verma, Abhishek, et al. [17] proposed utilizing machine learning classification techniques to protect IoT against DoS assaults. Seven machine learning classification techniques are evaluated for their performance. A random search strategy is used to identify the optimal features of classifiers. All classifiers are evaluated using the various IDS-based datasets. In addition, Friedman and Nemenyi post-host tests are applied to the statistical analysis of performance measurements to identify significant differences amongst classifiers. In addition, the mean response time of each classifier is assessed using a hardware device. Based on the practical outcome and statistical analysis, the XGBOOST provides the optimal balance amid prominent metrics and response time in IDS-based Internet of Things-specific anomaly. The next objective is to create an Intrusion Detection System for attacking IoT networks from routing assaults.

Faysal, Jabed Al, et al. [18] suggested a hybrid machine learning technique for identifying intrusion threats termed XGB-RF. This recommended paper investigated five distinct approaches. It has been demonstrated that XGB-RF was superior to other techniques in accuracy, sensitivity, and Kappa index. The effectiveness was assessed with more than 99 per cent accuracy on the N-BaIoT dataset, which is also greater than other conventional machine learning methods. This suggested method can significantly improve the safety of IoT systems because IoT device security and privacy are essential to their success. However, detecting 383,379 unknown attacks has proven challenging due to the rise in inventive attack types. Currently, the presented method for determining a single assault requires 0.0010063 seconds. It could eventually be implemented in a dynamic IoT system due to its ability to minimize detection time while maintaining high precision. They will also evaluate the machine learning classifiers in identifying unidentified attacks in IoT settings. Table.1 describes the literature review analysis.

Table.1. Literature Review Analysis

| S. No | Author | Model | Advantage | Disadvantage |
|---|---|---|---|---|
| 1 | Sarker, Iqbal H., et al. | IntruDTree machine-learning [9] | Increase Accuracy | Low effectiveness of collecting large datasets |
| 2 | Almseidin, Mohammad, et al. | Various machine learning classifiers [10] | High Accuracy | False negative and false favourable rates are not considered. |
| 3 | Alqahtani, Hamed, et al. | Account for popular classification techniques in machine learning [11] | Improve overall accuracy & security | It is not provided automated security services |
| 4 | Jamadar, Riyazahmed et al. | Decision tree machine learning algorithm[12] | High detection rate, High accuracy (99.9%)<br>and low False-Positive-Rate. | not considered in real-time packets |
| 5 | Bertoli, Gustavo De Carvalho, et al. | AB-TRAP framework [13] | Reproducible, Reduce network traffic & attacks. | Not considered Power Consumption |
| 6 | Alhajjar, Elie, et al. | Adversarial machine learning technique [14] | high misclassification rates & Low attacks | Error occur |
| 7 | Alzahrani, Abdulsalam O et al. | Decision Tree, Random Forest and XG Boost Machine Learning techniques [15] | High Accuracy | High Consumption of time |
| 8 | Megantara, Achmad Akbar, et al. | Hybrid machine learning method [16] | High Accuracy & Reduce the processing time | cut-off value |
| 9 | Verma, Abhishek, et al. | Machine Learning Classification Algorithm [17] | Reduce DOS Attacks | Chance of Error occurs |
| 10 | Faysal, Jabed Al, et al. | hybrid machine learning scheme (XGB-RF) [18] | Accuracy, Reduce Attacks in IOT networks. | High detection time |

### 2.1. Challenges in Existing Method

The frequency and severity of cyberattacks have increased dramatically along with the rapid evaluation of global computer network applications. The System requirements include dependability, reliability, and security. In this manner, the network must be continuously monitored and examined for indications of vulnerabilities and active attacks to detect intrusions. When performed appropriately, intrusion detection can quickly and accurately detect attacks that threaten a computer network's integrity. The problem of class imbalance has impacted classification performance while detecting network intrusions in real-time. Incomplete sampling can lead to losing important data, affecting the ability to identify the attacks. As a result, a robust detection method for attacks must be recommended. One central area for improvement in current IDS solutions is the requirement to increase the accuracy of IDS attack detection.

### 2.2. Contribution to overcome the Challenges

The detection and recognition module identifies different attacks by identifying instances of malice, which is essential in IDS. The proposed Deep Light Boltzmann Boosting Algorithm monitors network components to detect anomalous behaviour and misuse. This proposed model combines the Light Gradient Boosting Algorithm with the Deep Boltzmann Machine to enhance the detection approach in IDS attacks. Here, the Light GBM optimizes the DBM learning of parameters to promote the attack's identifications with low error and high attack detection accuracy.

## 3. Proposed Methodology and Implementation

The proposed framework will be executed to identify distinctive attacks such as Normal, Probe, U2R, R2L, and DoS. Identifying these systematic assaults will thus aid in the battle against various illegal activities.

A method used in machine learning and data analysis for feature extraction is called non-linear component analysis (NLCA). The goal of the NLCA approach is to identify non-linear patterns and correlations in the data. The goal of NLCA algorithms is to reduce the original high-dimensional data's dimensions while maintaining or strengthening the non-linear patterns beneath it. NLCA can simplify the representation and make it more suitable for visualization, classification, or other downstream tasks by lowering the dimensionality of the data. A dataset with several characteristics or variables is necessary for NLCA algorithms. Preprocessing the data may be necessary, including eliminating outliers, scaling or normalizing the features, and managing missing values.

In order to identify the non-linear correlations in the data, NLCA algorithms modify the data in a non-linear way. Numerous mathematical methods, including kernel functions, neural networks, and manifold learning algorithms, can be used to produce this change. After that, a lower-dimensional space is projected using the altered data. The intended level of dimensionality reduction or the amount of information preservation needed are often taken into consideration when determining the number of dimensions in the reduced space. The retrieved features in NLCA are thought of as the lower-dimensional representation that results from dimensionality reduction. The significant non-linear patterns or structures seen in the original data should be preserved by these characteristics.

The retrieved features might be utilized for more research, visualization, or as a starting point for later machine learning tasks like regression, classification, or clustering. When working with complicated, high-dimensional datasets that show non-linear correlations between variables, NLCA is very helpful. A more effective analysis and decision-making are made possible by NLCA by revealing hidden patterns, separating classes or clusters, and providing a more comprehensible representation of the data through the extraction of non-linear characteristics.

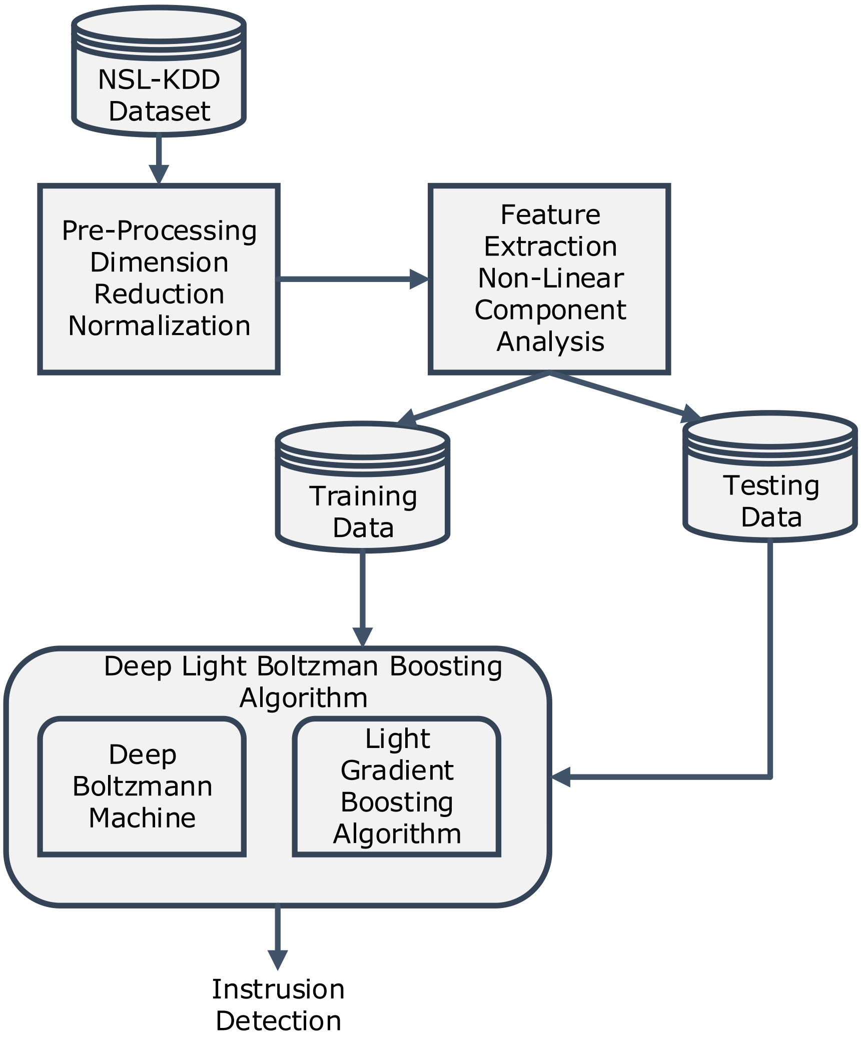

NSL-KDD Dataset
Pre-Processing Dimension Reduction Normalization
Feature Extraction Non-Linear Component Analysis
Training Data
Testing Data
Deep Light Boltzman Boosting Algorithm
Deep Boltzmann Machine
Light Gradient Boosting Algorithm
Instrusion Detection

**Fig.1. Proposed Methodology**

Initially, the dataset is entered into the preprocessing stage, so the noise and redundant data may be removed by dimension reduction and normalization approach [19]. To improve the classification method, features are retrieved using a method called nonlinear component analysis[20]. In this case, feature extraction can help select particular variables as well as combine some of the related variables, which would reduce the amount of data. According to it, nonlinear techniques enable statistical redundancy removal without information loss and provide more reliable nonlinear pattern discovery for high-dimensional data. After that, the datasets are divided into training and testing datasets. The proposed novel classifier is trained using the newly proposed training data. The light Gradient boosting algorithm is used to perfect the hidden layers of the deep Boltzmann distribution here. Boosting with gradients is a form of ensemble machine learning that has found widespread application in data science to address classification and regression issues. It is simple to use and is compatible with many different types of data, including those that are small and heterogeneous. In the light Gradient boosting Algorithm ensemble model, these Gradient boosted decision trees are used. Throughout the training, a series of decision trees are gradually constructed. The subsequent trees are built with a lesser level of less than the ones that came before them. As a result, there is a decrease in inaccuracy that occurs throughout the classification. As a direct consequence of this, attacks are successfully classified.

### 3.1. Proposed Light Gradient Boosting Approach with Deep Boltzmann Machine

The parameters of the Light method are optimized with the help of a Boltzmann Machine (BM)-based hyperparameter optimization algorithm. Light Gradient boosting algorithm that can swiftly analyze massive datasets and handle distributed data processing [21]. Several controls, known as hyperparameters, are available in the Light gradient algorithm. Light gradient algorithm performance is susceptible to hyperparameters. To fine-tune the hyperparameters of the optimization algorithm, this study integrates a Bayesian-based hyperparameter Light gradient algorithm into a Deep Boltzmann Machine (BM) based approach. In this case, the Deep Light Boltzmann Boosting Technique uses a refined version of the Light GBM algorithm. During the structural design of a neural network, this optimization algorithm is used to determine the optimum number of units on the input layer and the hidden layer, as well as the optimum learning rate, with the end goal of enhancing the accuracy of predictions for mechanical management in manufacturing. The "num leaves" parameter, which represents the number of leaves in a tree, the "max depth" parameter, which means the maximum depth of a tree, and the "learning rate" parameter are all hyperparameters that can be fine-tuned.

A Boltzmann Machine (BM) [22] is a network in which some nodes are exposed while others remain concealed; these nodes are symmetrically coupled. Multiple hidden layers and one exposed one make up a deep Boltzmann machine. The correlations between the units in successive layers are captured at higher and higher orders. It will review the procedures involved in teaching a DBM to work with either a single or more traditional modality. Consider a DBM with overt Gaussian units and obtrusive binary ones.

The DBM's ground state energy is therefore calculated as equation (1)

$$F\left(\frac{m}{\theta}\right) = \sum_{a=1}^{k} \quad \frac{(U_a - Y_a)^2}{2\sigma_a^2} - \sum_{a-1}^{k} \quad \sum_{b-1}^{R_1} \quad \frac{U_a}{\sigma_a^2} E \frac{1}{ab} l_b^1 - \sum_{a=1}^{2} \quad \sum_{b=1}^{R_n} \quad G_b^n l_b^n - \sum_{b=1}^{R_1} \quad \sum_{t=1}^{R_2} \quad l_b^1 E_{bt}^2 l_{t,}^2 \; (1)$$

$Y_a$ and $G_b^1$, respectively, indicate the biases of the units of visible and first hidden layer; additionally, the visible-to-hidden and hidden-to-hidden units based on the symmetric interaction terms are obtained using equations (2-3)

$$E^1 = \left\{E_{a,b}^1\right\} \tag{2}$$

$$E^2 = \left\{E_{b,t}^2\right\} \tag{3}$$

If the set of variables is expressed as$\theta = \{E^1, E^2\}$ based on the energy function, DBM employs a Boltzmann distribution to assign a probability to the state vector as below,

$$R\left(\frac{m}{\theta}\right) = \frac{1}{\omega(\theta)} exp\left(-W\left(\frac{m}{\theta}\right)\right), \tag{4}$$

Where $\omega(\theta)$indicates partition function. By maximizing the log-likelihood, the parameters can be learned,

$$V = \sum_{z=1}^{A} \quad log \sum_{l^1,l^2} \quad R\left(\mu^{(z)}\right), \frac{l}{\theta}(5)$$

Given A training samples $\{\mu^z\}_{z=1,\ldots,A}$and $l = \{l^1, l^2\}$. The Gradient, which is produced using the partial derivative of the log-likelihood function, is used to update the DBMparamter once every mini-batch:

$$\frac{\vartheta v\left(\frac{\theta}{\mu}\right)}{\vartheta\theta} = \langle\frac{\vartheta W\left(\mu^z, \frac{l}{\theta}\right)}{\vartheta\theta}\rangle_i - \langle\frac{\vartheta W\left(\mu, \frac{l}{\theta}\right)}{\vartheta\theta}\rangle_p \tag{6}$$

Where $\langle . \rangle_i$and $\langle . \rangle_p$ indicate the expectations of the data distribution $R\left(\frac{l}{\{\mu^z\}} \theta\right)$and the model distribution $R \left(\mu, \frac{l}{\theta}\right)$respectively. The updating principles are well described. However, accurate computation is challenging.

The variational parameters for the n-th layer in the unit,$\rho_\alpha^1$are evaluated by:

$$\rho_\alpha^1 \leftarrow \sigma\left(\sum_{a=1}^{R_{l-1}} \quad \rho_a^{l-1} + \sum_{t=1}^{R_{l+1}} \quad \rho_t^{l+1} E_{\alpha t}^l + G_b^l\right) \tag{7}$$

In terms of the current parameters, this technique returns variational parameter values that maximize the following lower bound:

$$\delta(\mu; \theta) \geq W_{T(l)}[-W(v, l)] + C(T) - log\; log\; x(\theta) \tag{8}$$

The entropy function as follow:

$$C(T) = -\sum_{a=1}^{2} \sum_{b=1}^{R_l} \left(\rho_b^l log \rho_b^l + (1-\rho_b^l) log(1-\rho_b^l)\right) \quad (9)$$

The joint probability distribution over s and f can be described if unit 7 consists of the units shared by the unimodal DBMs.

$$R\left(x, \frac{y}{\theta}\right) = \sum_{l_x^2, l_y^2, l^3} R(l_x^2, l_y^2, l^3) \left(\sum_{l_x^1} R(x, l_x^1, l_x^2)\right)\left(\sum_{l_y^1} R(y, l_y^1, l_y^1)\right) \quad (10)$$

The parameter is set, where the real posterior$\theta = \{E_x^1,\ E_x^2, E_x,\ E_y^1,\ E_y^2,\ E_y\}$is approximated using a fully factorized approximation distribution.

$$C\left(\frac{l}{x}, y; \rho\right) = \left(C \prod_{b=1}^{R_1} c\left(\frac{l_{xb}^1}{x,y}\right) \prod_{t=1}^{R_2} c\left(\frac{l_{xt}^2}{x,y}\right)\right)\left(C \prod_{b=1}^{R_1} c\left(\frac{l_{yb}^1}{x,y}\right) \prod_{t=1}^{R_2} c\left(\frac{l_{yt}^2}{x,y}\right)\left(C \prod_{M=1}^{R_3} c\left(\frac{l_M^1}{x,y}\right)\right)\right)$$

(11)

Where $l = \{l_x^1, l_x^2, l_y^1, l_y^2, l^3\}$ and $\rho = \{\rho_x^1, \rho_x^2, \rho_y^1, \rho_y^2, \rho^3\}$ are the variational parameters.

These accurate posterior parameters $\theta$are optimized Light Gradient Boosting Algorithm for optimal parameter Determination. Gradient boosting decision tree (GBDT) is a well-known and frequently performed approach for regression and classification issues. The GBDT algorithm is named LightGBM, which is a combination of gradient-based one-sided sampling (GOSS) and Exclusive Feature Bundling (EFB) to process large numbers [23]. Data samples, as well as a large number of corresponding features. To improve training, LGBM uses a Gradient-based Single Sided Sampling (GOSS) sampling algorithm to indicate the significance of data versions. Data with small gradients have fewer errors, which suggests that they were adequately trained, according to the premise.GOSS recommends ignoring these less informative data points and using the rest to calculate the information obtained when searching for the best division. Besides, LGBM uses an Exclusive Feature Bundling algorithm to deal with dataset scarcity. It effectively merges mutually exclusive characteristics, producing fewer features while maintaining the most data.

Combine features that never accept nonzero values simultaneously of $a^i, i = \{1, \dots \dots, N\}$ using the exclusive features bundling (EFB) technique. The optimization objective of GBDT is to minimize the loss function, as below in (12),

$Set\ \theta_0(a) = arg\ arg\ min_E \sum_m^N W(b_n, E)$;(12)

For GBDT, all data versions must be analyzed to calculate the knowledge gain of each feasible performance. GOSS accelerates this technique by eliminating the majority of cases with minor gradients. The training cases should be first categorized by the optimum value of the Gradient represented in descending order. in equation (13)

$$t_n = \left\|\frac{\vartheta W(b_n, \theta(a_n))}{\vartheta \theta(a_n)}\right\|_{\theta(a)=\theta_{m-1(a)}} \quad , n= \{1,\dots.. N\} \quad (13)$$

Afterwards, cases are divided by information gained according to equation (14)

$$U_m(q) = \frac{1}{n}\left(\left(\frac{\sum \quad an \in_{X_f} t_n + \frac{1-x}{y} \sum \quad an \in_{Y_f} t_n}{n_f^m(q)}\right)^2 + \left(\frac{\sum \quad an \in_{X_t} t_n + \frac{1-x}{y} \sum \quad an \in_{Y_t} t_n}{n_t^m(q)}\right)^2\right) \quad (14)$$

Learning is accomplished by solving a system of mean-field equations searching for the value that increases the differentiated lower range for the simulation parameter's present value. As a result, the light Gradient boosting classifier improves the weak classifier of DBM and promotes the classification of attacks in IDS with less feature dimension, error and high accuracy. The proposed Light Gradient Boosting approach with Deep Boltzmann Machine Algorithm is shown in Algorithm 1.

Algorithm.1. Pseudocode of Light Gradient Boosting approach with Deep Boltzmann Machine

```
Step 1: Start
Step 2: The input of IDS data is trained using the DBM method
Step 3: The training algorithm is initiated
Step 4: Learn to share layer parameters
Step 5: Calculate the entropy function
Step 6: Define joint probability distribution for shared accurate layer information
Step 7: Calculate the approximate posteriors of the unimodal DBMs
Step 8: Initialize the posteriors parameter for tuning by LGBM
Step 9: Compute the absolute values of the Gradient
Step 10: Resample data set using Gradient-based one-side sampling (GOSS) process
top N= x= len(Q); randN= y=len(Q);
sorted= Get sorted Indices (xyP(t));
X= sort Out [1-top N]; Y= Random pick (sort out [top N: len(Q)], and N);
Step 11: Calculate the gain of information
Step 12:Develop a novel decision tree (DT)$\theta_s(a)'$ set on $Q'$
Step 13: Update $\theta_{s-1}(a) + \theta_s(a)$
```

## 4. Result and Discussion

### 4.1 Dataset

Regarding data mining competitions, the KDD Cup stands head and shoulders above the rest of the pack. Many problems seen in the KDD Cup 1999 dataset were intended to be addressed by the NSL-KDD dataset. Numerous researchers have utilized this dataset to test and develop the NIDS problem. The collection contains assaults of every variety. The dataset comprises forty-one features, divided into ordinary and attack labels and three primary kinds (essential features, according to content and traffic features). There are four distinct types of classifications. Table 2 gives an overview of the many forms of attack and their damage.

Table.2 Description of Types of Attacks

| Types of Attacks | Description |
|---|---|
| DoS | Denial of Service for Cyber Attack |
| R2L | Remote to User for computer-based attack |
| U2R | User to Root Attack for remote-based computer attack |
| Probe | Cyber Attack |

### 4.2. Comparison of Proposed Model with Existing Models

Experiments have been published for building the Intrusion detection system to identify intrusions. The predicted classification accuracy specifics using the suggested model and the Deep Light Boltzmann Boosting Algorithm are provided here. An NSL-KDD dataset is used in the implementation, and the Deep Light Boltzmann Boosting Algorithm is used to assess the results. When using NSL-KDD, data is automatically split into a training and evaluation phase. Accordingly, the processed data are sent into the algorithm, where they are identified as either an attack or expected behaviour. The latter is further analyzed to determine its category, and the former is acted upon accordingly. The suggested model is evaluated by comparing it to other models already used for performance analysis. Here, the suggested Deep Light Boltzmann Boosting Algorithm is contrasted with some intrusion detection-based, well-establishedconventionalapproches, including the K-means Neural Network (KNN) [24], the Decision Tree (DT) [25], SVM [26], and Support Vector Machine with XGBOOST (SVMXG) [27] were utilized. In addition, the parameters from the confusion matrix can be used to determine true positive, false positive, true negative, and false negative success rates for detecting and classifying anomalous instances, regular instances, and false positives. The confusion matrix-based metrics value is obtained using the proposed system, and conventional models are displayed in Table 3.

Table. 3. Comparison of Instruction detection

| Attacks | Methodologies | Precision | Recall | Specificity | F-measure | Accuracy |
|---|---|---|---|---|---|---|
| DoS | K-means Neural Network (KNN) [24] | 85.5 | 84.8 | 83.5. | 86.8 | 99.7 |
| | Decision Tree Classifier (DT) [25] | 74.6 | 78.56 | 72.67 | 76.8 | 78.23 |
| | Support Vector Machine (SVM) [26] | 82.1 | 79.3 | 80.2 | 78.5 | 83.6 |
| | Support Vector Machine with XGBOOST (SVMXG) [27] | 88.6 | 85.6 | 89.78 | 86.7 | 90.5 |
| | Light Gradient Boosting Approach with Deep Boltzmann Machine (Proposed System) | 97.89 | 99 | 96.78 | 98.44 | 99.7 |
| R2L | K-means Neural Network (KNN) [24] | 95.5 | 96.8 | 95.5. | 96.6 | 94.1 |
| | Decision Tree Classifier (DT) [25] | 84.6 | 88.56 | 92.67 | 86.8 | 90.23 |
| | Support Vector Machine (SVM) [26] | 88.1 | 78.3 | 81.2 | 77.5 | 87.8 |
| | Support Vector Machine with XGBOOST (SVMXG) [27] | 98.6 | 95.6 | 85.78 | 96.7 | 95.5 |
| | Light Gradient Boosting Approach with Deep Boltzmann Machine (Proposed System) | 96.8 | 97.34 | 92.49 | 97.06 | 98.6 |
| U2R | K-means Neural Network (KNN) [24] | 78.3 | 76.6 | 75.8 | 76.4 | 80.7 |
| | Decision Tree Classifier (DT) [25] | 85.7 | 87.56 | 93.67 | 96.8 | 91.13 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | Support Vector Machine (SVM) [26] | 89.1 | 79.3 | 89.2 | 79.5 | 89.8 |
| | Support Vector Machine with XGBOOST (SVMXG) [27] | 95.8 | 94.9 | 95.8 | 96.9 | 96.7 |
| | Light Gradient Boosting Approach with Deep Boltzmann Machine (Proposed System) | 93.33 | 91.2 | 95.6 | 92.25 | 93.5 |
| Probe | K-means Neural Network (KNN) [24] | 98.3 | 96.6 | 95.8 | 97.4 | 97.7 |
| | Decision Tree Classifier (DT) [25] | 95.7 | 97.56 | 95.67 | 96.8 | 97.13 |
| | Support Vector Machine (SVM) [26] | 99.1 | 99.2 | 98.2 | 93.5 | 99.8 |
| | Support Vector Machine with XGBOOST (SVMXG) [27] | 96.8 | 97.9 | 98.8 | 97.9 | 99.7 |
| | Light Gradient Boosting Approach with Deep Boltzmann Machine (Proposed System) | 95.6 | 97.5 | 93.21 | 96.54 | 96.32 |
| Normal | K-means Neural Network (KNN) [24] | 98.3 | 96.6 | 95.8 | 97.4 | 97.7 |
| | Decision Tree Classifier (DT) [25] | 95.7 | 97.56 | 95.67 | 96.8 | 97.13 |
| | Support Vector Machine (SVM) [26] | 99.1 | 99.2 | 98.2 | 93.5 | 99.8 |
| | Support Vector Machine with XGBOOST (SVMXG) [27] | 96.8 | 97.9 | 98.8 | 97.9 | 99.7 |
| | Light Gradient Boosting Approach with Deep Boltzmann Machine (Proposed System) | 98.3 | 99.2 | 98.1 | 98.69 | 98.4 |

As shown in Table 3, evaluating the Deep Light Boltzmann Boosting algorithm model categorization as usual and attacks acquired from the confusion metrics revealed a better performance. In addition, we compare them to DoS, R2L, U2R, Probe, and Normal in terms of accuracies, recalls, Specificities, F1 scores, and precisions. In general, the provided algorithms successfully detected assaults.Fig.2 displays the resulting DoS attacks analysis as below,

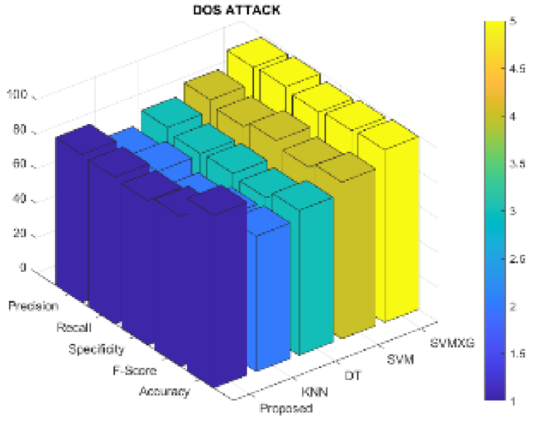


Fig.2. Comparison of DoS Attacks in Different Classifiers.

As shown in Fig. 2, the suggested model employing the Deep Light Boltzmann Boosting method achieves higher precision, Recall, Specificity, F-measure and Accuracy in detecting Dos attacks. K-means Neural Network (KNN), Decision Tree Classifier (DT), Support Vector Machine (SVM), and Support Vector Machine with XGBOOST all performed worse than the suggested model, which achieved 97.89%, 99%, 96.7%, 98.44%, 99.7%, respectively (SVMXG). The results of this effort to recognize R2L attacks are then graphically depicted in Fig. 3.

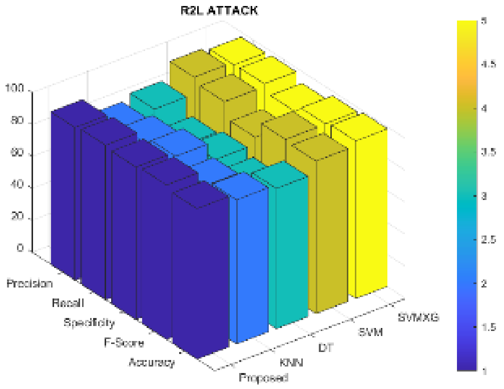


Fig.3. Comparison of R2L Attack in different Classifiers.

As shown, Fig.3 offers a detailed demonstration of attack detection results achieved by various methods. Most machine learning approaches listed are intended to detect network intrusions and malware. It chooses metrics as evaluations from the list of metrics because most of the mentioned methods employ these values for experiments

.Deep Light Boltzmann Boosting Algorithm (Proposed) may perform better than existing machine learning approaches. The outstanding performance obtained byU2R attack detection methods on the NSL-KDD dataset, that is, 93.5% in terms of accuracy for the proposed, 96.7% achieved by SVMXG, which proves that SVMXG slightly higher than the proposed model is shown in Fig.4

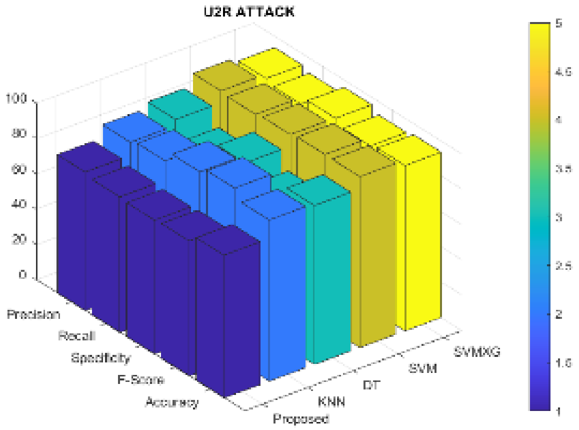


Fig.4. Comparison of R2L Attack in different Classifiers.

Figure 4 shows that the suggested model achieves a higher level of accuracy more quickly than the KNN, DT, and SVM methods. A further circumstance is that XGBOOST provides marginally better accuracy than the Support Vector Machine (SVMXG). Fig. 5 shows the Probe's attack detection results.

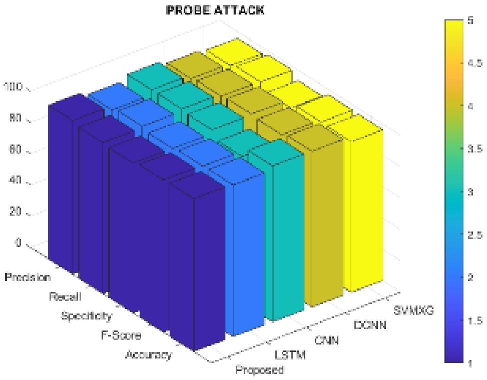
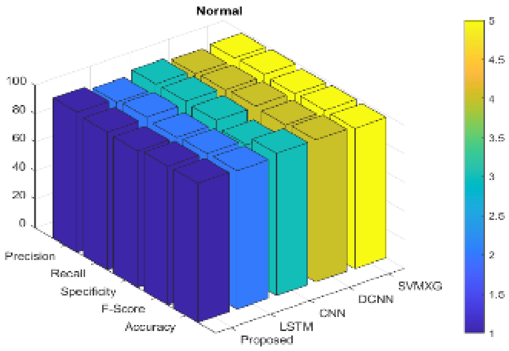


Fig.5. Comparison of Probe Attacks in different Classifiers

In fig.5 analyzed the probing attack detection rates with conventional ML classifiers. With the anomaly detector of the proposed model getting high values, other approaches get low values in metrics of accuracy, recall, specificity, F-measure, and precision.

According to Table.4 for regular or non-attack data, the attack detection with the proposed model outperforms conventional approaches, as shown in Fig.6,



.

Fig.6. Comparison of Normal Attack in different Classifiers

Figure 6 displays the impressive results of the Deep Light Boltzmann Boosting Algorithm. The system achieves an outstanding 98.3% precision, 99.2% recall, 98.1% specificity, 98.69% F-measure, and 98.4% accuracy. Increasing the number of machine learning classifiers would likely improve detection performance.

## 5. Conclusion

The security will be crucial in future networks such as IoT, Cloud Computing, Sensor networks, etc. However, many security techniques designed to safeguard the security of such networks from assaults are challenging to execute because of the limited capacity of networking devices. The IDS method is applied to the given model, and the assaults are detected in real-time. In this model, the Light Gradient Boosting technique was employed with the Deep Boltzmann Machine Classifier to form an ensemble model. The error rate in the deep learning classifier is decreased, thanks to the Gradient boosting techniques. In a wireless sensor network, the system can identify a flooding attack. Studies have also been done to calculate the impact of several attacks (DoS, Probe, U2R, R2L, and Normal). Attack and non-attack detection analysis showed that the suggested model performs well over conventional approaches by wide margins for DoS, Probe, U2R, R2L, and Normal:99.7%, 96.32%, 93.5%, 98.6%, and 98.4%, respectively. In conclusion, the discussed techniques have shown promise for more automated attainment of accuracy levels.

It is suggested to use feature extraction and feature selection as a hybrid strategy to enhance the accuracy of intrusion detection using large datasets through autoencoder and RNN-based methods that can be combined in models to boost accuracy as our future enhancement.

## Reference

[1] Sumra, Irshad Ahmed, Halabi Bin Hasbullah, and Jamalul-lail Bin AbManan. "Attacks on security goals (confidentiality, integrity, availability) in VANET: a survey." Vehicular Ad-Hoc Networks for Smart Cities. Springer, Singapore, 2015. 51-61.

[2] Ghaben, Ayman, et al. "Mathematical Approach as Qualitative Metrics of Distributed Denial of Service Attack Detection Mechanisms." IEEE Access 9 (2021): 123012-123028.

[3] Nazir, Anjum, and Rizwan Ahmed Khan. "Network intrusion detection: taxonomy and machine learning applications." Machine intelligence and big data analytics for cybersecurity applications. Springer, Cham, 2021. 3-28.

[4] Manhas, Jatinder, and Shallu Kotwal. "Implementation of intrusion detection system for internet of things using machine learning techniques." Multimedia Security. Springer, Singapore, 2021. 217-237.

[5] Alzahrani, Rami J., and Ahmed Alzahrani. "Security Analysis of DDoS Attacks Using Machine Learning Algorithms in Networks Traffic." Electronics 10.23 (2021): 2919.

[6]Zhu, Guangyi. "Automated False Positive Filtering for network Alerts." arXiv preprint arXiv:2208.12729 (2022).

[7]Dobrovska, Lyudmila, and Olena Nosovets. "Development of the Classifier Based on a Multilayer Perceptron Using Genetic Algorithm and Cart Decision Tree." Eastern-European Journal of Enterprise Technologies 5.9 (2021): 113.

[8]Yu, Keping, et al. "Securing critical infrastructures: deep-learning-based threat detection in IIoT." IEEE Communications Magazine 59.10 (2021): 76-82.

[9].Sarker, Iqbal H., et al. "Intrudtree: a machine learning based cyber security intrusion detection model." *Symmetry* 12.5 (2020): 754.

[10]. Almseidin, Mohammad, et al. "Evaluation of machine learning algorithms for the intrusion detection system." *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*. IEEE, 2017.

[11]. Alqahtani, Hamed, et al. "Cyber intrusion detection using machine learning classification techniques." *International conference on computing science, communication and security*. Springer, Singapore, 2020.

[12]. Jamadar, Riyazahmed A. "Network intrusion detection system using machine learning." *Indian Journal of Science and Technology* 7.48 (2018): 1-6.

[13]. Bertoli, Gustavo De Carvalho, et al. "An end-to-end framework for machine learning-based network intrusion detection system." *IEEE Access* 9 (2021): 106790-106805.

[14]. Alhajjar, Elie, Paul Maxwell, and Nathaniel Bastian. "Adversarial machine learning in network intrusion detection systems." *Expert Systems with Applications* 186 (2021): 115782.

[15]. Alzahrani, Abdulsalam O., and Mohammed JF Alenazi. "Designing a network intrusion detection system based on machine learning for software-defined networks." *Future Internet* 13.5 (2021): 111.

[16]. Megantara, Achmad Akbar, and Tohari Ahmad. "A hybrid machine learning method for increasing the performance of network intrusion detection systems." *Journal of Big Data* 8.1 (2021): 1-19.

[17] Verma, Abhishek, and Virender Ranga. "Machine learning based intrusion detection systems for IoT applications." *Wireless Personal Communications* 111.4 (2020): 2287-2310.

[18] Faysal, Jabed Al, et al. "XGB-RF: A hybrid machine learning approach for IoT intrusion detection." *Telecom*. Vol. 3. No. 1. MDPI, 2022.

[19]McCarthy, Davis J., et al. "Scater: preprocessing, quality control, normalization and visualization of single-cell RNA-seq data in R." Bioinformatics 33.8 (2017): 1179-1186.

[20]Jiménez, Alfredo Arcos, et al. "Linear and nonlinear features and machine learning for wind turbine blade ice detection and diagnosis." Renewable energy 132 (2019): 1034-1048.

[21]Kumar, Sanjay, Abhishek Mallik, and B. S. Panda. "Link prediction in complex networks using node centrality and light gradient boosting machine." World Wide Web 25.6 (2022): 2487-2513.

[22]Fissore, Giancarlo. Generative modelling: statistical physics of Restricted Boltzmann Machines, learning with missing information and scalable training of Linear Flows. Diss. université Paris-Saclay, 2022.

[23] Jiang, Jun, et al. "A dynamic ensemble algorithm for anomaly detection in IoT imbalanced data streams." Computer Communications 194 (2022): 250-257.

[24]Ding, Hongwei, et al. "Imbalanced data classification: A KNN and generative adversarial networks-based hybrid approach for intrusion detection." Future Generation Computer Systems 131 (2022): 240-254.

[25]Bagui, Sikha, et al. "Spark configurations to optimize decision tree classification on UNSW-NB15." Big Data and Cognitive Computing 6.2 (2022): 38.

[26]Ponmalar, A., and V. Dhanakoti. "An intrusion detection approach using ensemble support vector machine based chaos game optimization algorithm in the big data platform." Applied Soft Computing 116 (2022): 108295.

[27]Fridayanthie, Eka Wulansari. "Optimization of Support Vector Machine and XGBoost Methods Using Feature Selection to Improve Classification Performance." JOURNAL OF INFORMATICS AND TELECOMMUNICATION ENGINEERING 6.2 (2023): 484-493.