STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **14**, July 2025, pp 434–453.
Published online in International Academic Press (www.IAPress.org)

Stacked Ensemble Method: An Advanced Machine Learning Approach for Anomaly-based Intrusion Detection System

Anichur Rahman[1,2,*], Md. Saikat Islam Khan[2], MD. Zunead Abedin Eidmum[3], Pabon Shaha[2], Bakhtiar Muiz[3], Nahid Hasan[4], Tanoy Debnath[2], Dipanjali Kundu[1], Jarin Tasnim Tamanna[1], Mohammad Sayduzzaman[1], and Muaz Rahman[5]

[1]*Department of Computer Science and Engineering, National Institute of Textile Engineering and Research (NITER), Constituent Institute of the University of Dhaka, Savar, Dhaka-1350, Bangladesh*

[2]*Department of CSE, Mawlana Bhashani Science and Technology University, Tangail, Bangladesh*

[3]*Department of Internet of Things and Robotics Engineering, Bangabandhu Sheikh Mujibur Rahman Digital University, Bangladesh*

[4]*Department of Computer Science and Engineering, Daffodil International University, Savar, Dhaka, Bangladesh*

[5]*Department of Electrical and Electronic Engineering, National Institute of Textile Engineering and Research (NITER), Constituent Institute of the University of Dhaka, Savar, Dhaka-1350, Bangladesh*

Abstract The subject of this article is IDS-Intrusion Detection Systems, which are strongly related to a comprehensive cyber attack prevention system. In the present day, an IDS for network infrastructure is a crucial topic. The advancement of SDN-Software Defined Networking has led to a rising need for software-based IDS-Intrusion Detection Systems. Diverse methodologies, including machine learning algorithms and other statistical models, have been used to develop distinct kinds of IDS- Intrusion Detection Systems to enhance performance and opportunity still exists to improve further. Several studies have focused on solving these problems for this reason, utilizing methods like conventional machine learning models. However, existing systems need to improve, including a low detection rate and a high false alarm rate. The aim is to improve performance, specifically in terms of increases in detection rate. This work introduces a new IDS-Intrusion Detection System named SIDS-Stacked Intrusion Detection System, which utilizes a stack-based approach to improve detection accuracy and resilience. The objective is to utilize various predictive algorithms most efficiently. An ensemble classifier method is used to enhance the precision of the final prediction by amalgamating the outputs of multiple models. This research implemented numerous ML-machine learning methodologies, including Stochastic Gradient Descent, Logistic Regression, Random Forest, and Deep Neural Networks, to construct a multilayered model that would optimize network intrusion detection accuracy. This challenging research project employs the NSL-KDD dataset. In previous studies, the stacked model (DNN1 + DNN2) has a maximum accuracy of 97.90% for intrusion detection. However, the suggested trained model outperforms existing models by 99.98%. Additionally, the offered stacked model attains F1-score 99.2%, a FPR-false positive rate 4.4%, and a FNR-false negative rate 0.18%. In conclusion, the findings indicate that a stacked ensemble method can enhance evaluation metrics and provide more consistent performance. We made all the materials publicly accessible for the research community. They can be retrieved from: https://github.com/jarin188/NSL-KDD.

Keywords IDS-Intrusion Detection System, SIDS-Stacked Intrusion Detection System, NSL-KDD, Machine Learning, Deep Learning, Stack Model, DataSets, Data Analysis.

AMS 2010 subject classifications 97P40, 97P60, 97R40

DOI: 10.19139/soic-2310-5070-2352

[*]Correspondence to: Anichur Rahman (Email: anis_cse@niter.edu.bd). Department of CSE, National Institute of Textile Engineering and Research (NITER), Constituent Institute of the University of Dhaka, Savar, Dhaka-1350, Bangladesh.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

1. Introduction

Presently, Internet usage has dramatically increased since a large amount of sensitive data is stored online. So, security is an important issue to protect the network system from external attacks [1]. Security attackers attack a networking system in various ways. IDS is meant to find out when someone is doing something wrong or illegal on a computer network. An IDS system is hardware that includes software elements and automates intrusion detection. IDS helps detect intrusion attacks by studying a few knowledge documents used in the network types. Besides, the failure to mitigate intrusions may undermine the reliability of network security services, such as data protection, transparency, and availability. Intrusion detection systems (IDS) are essential to cybersecurity because they can identify security threats and criminal activity. IDS solutions assist organisations in preventing data breaches, unauthorised access, and network invasions due to the growing complexity and sophistication of cyberattacks. Thus, this brought about the need for intrusion detection because of the confidentiality of the communications within the various utilized networks. Several critical factors can influence stacked ensemble methods within an IDS-intrusion detection system. There are several works based on IDS system, most of which require extensive computational resources and excessive false alarms making them inefficient for real-time applications. Using advanced machine learning methods like stacked ensembles can make it work much better, which is what this study was all about. Emerged with different techniques and predicting algorithms, this study implemented a unique stake model to predict the intrusion of the network, which performs in a diverse way [2].

Table 1. List of Acronyms

Abbreviation	**Description**
AI	Artificial Intelligence
DDoS	Distributed Denial of Service
DST-TL	Deep Self Taught based Transfer Learning
DT	Decision Table
IDS	Intrusion Detection System
IoT	Internet of Things
LR	Logistic Regression
ML	machine Learning
ML-IDP	Machine Learning based Intelligent Document Processing
MLP	Multi Layer Perceptron
PART	Projective Adaptive Resonance Theory
RBFN	Radial Basis Function Network
RF	Random Forest
SIDS	Stacked IDS
SDN	Software Defined Networking
SGD	Stochastic Gradient Descent

These days, new networking technologies like SDN-Software Defined Networking, Blockchain, and the IoT-Internet of Things are helping to build new network systems [3]. Numerous applications exist for research into the development of blockchain-based SDN IoT networks [4]. These have garnered significant interest from scholars as a contemporary approach to network building. These dynamic technologies are broadly related to numerous network design infrastructures. Additionally, separation of the control plane from the data plane is facilitated by SDN and IoT technologies, which in turn improve network programmability. Nonetheless, the susceptibility to attacks is heightened with SDN (Table. 1) when Software-Defined Networking (SDN) defines its functional architecture as a control layer, data layer, and application layer. The authors in [5] also highlighted the issue of attack by external intruders in an intraorganization network on a shared resource platform. A network engineer usually deploys IDS to secure the network from potential intruders. ML approaches have a strong potential to design a dynamic IDS.

Furthermore, some researchers have suggested several machine learning-based attack detection methods to identify intrusions across a distributed network. For example, Random Forest can be the best option to gain high test accuracy, Multi-layered Artificial Intelligence (AI) - based protective mechanisms that can successfully detect intrusions in an SDN/Network Function Visualization (NFV)-enabled cloud of 5G networks. Furthermore, a number of researchers proposed the creation of an IDS that relies on deep neural networks. However, more methods than the ones stated above are needed to fully resolve the problem of detecting intrusions accurately within a network [6].

This work develops an accurate and efficient intrusion detection system called SIDS by employing specialized machine learning techniques on the standard NSL-KDD dataset. The NSL-KDD dataset is a recognized benchmark in examining intrusion detection techniques [7].

The researchers utilized a mean encoding technique. Later, a layered technique was used to improve the suggested layered Intrusion Detection System's (SIDS) stability. The studies showcase that SIDS attained a 98% accuracy, exceeding other similar techniques already in use. This study presents the following contributions:

- Researchers have used an ensemble technique to build a Stacked model for Stacked Intrusion Detection System (SIDS).
- Furthermore, used to build the stacked model are many ML-machine learning techniques like LR-Logistic Regression, RF-Random Forest, SGD-Stochastic Gradient Descent, and DNN-Deep Neural Network.
- The SIDS beats existing systems with an accuracy of 99.98% and f1-score 0.992.
- The SIDS also has a lower rate of false positives 0.0448 and a lower rate of false negatives 0.0018, making it better than the other models.
- Finally, the implementation of cross-validation to validate model performance ensures model stability and reliability across different subsets of the data.

To improve intrusion detection, data integrity, efficient network management, and regulatory compliance are the main objectives of this research. Therefore, the approaches considered include data collection and preprocessing, model creation integrated with blockchain, and real-time monitoring. The model proposed was supported by testing and evaluation, and, most importantly, security awareness.

The following sections of the paper are structured as outlined below: Background knowledge is provided along with relevant works in the section 2 and 3. Section 4 goes into more information about the suggested ML model. In sections 5 and 6, it is about the tests, their results, their limits, and what they mean for future work. Finally, this study's conclusions in section 7 explain how important it is and what it means for the future.

2. Background Study

This section describes the background and related studies regarding IDS and ML.

2.1. Intrusion Detection System (IDS)

An intruder is someone who deliberately enters the network infrastructure and conducts malicious operations, and the detection procedure is named intruder detection system (IDS). Researchers have proposed many ways to detect malicious attacks as well as attackers. In [8], a machine learning-based approach where features were selected based on a Bayesian classifier to identify intruders' attacks. In other similar research, deep reinforcement learning, decision trees, and deep neural network algorithms were applied to identify intruders in networking systems. However, some other approaches are also effective for this overall security management procedure, such as the Blockchain system [9]. In [10], with the help of information from the network manager about traffic routing, DDoS attacks can be found quickly. It was emphasised by the writers that SDN-based IDS can be used to find and stop DDoS attacks. The SIDS model may be vulnerable to various security threats such as adversarial attacks and poisoning attacks during training. Adversarial attacks can manipulate input data, leading the model to misclassify malicious traffic as benign, but this can be countered by incorporating adversarial examples during training, using regularization techniques, and implementing anomaly detection for perturbed inputs. Poisoning attacks, where malicious data is introduced during training, can be mitigated through data sanitization, robust learning algorithms, and model verification. Post-deployment, ensuring model integrity is critical, and this can be achieved through secure deployment methods like encryption, checksum verification, and monitoring for model drift. Privacy concerns can be addressed by using federated learning and differential privacy techniques to prevent data leakage. Lastly, evasion attacks, where attackers craft traffic that avoids detection, can be countered by regular updates, retraining with new data, and using hybrid detection methods combining various IDS techniques. These countermeasures collectively enhance the security, robustness, and integrity of the SIDS model, ensuring its effectiveness in real-world scenarios.

2.2. How Does Machine Learning Work?

Machine learning approaches are applied extensively in several fields. The protection of any application focuses primarily on preventing intrusions, detecting any form of phishing, preserving privacy, detecting spam, etc. [11]. A machine learning algorithm's main benefit is that it can learn from past examples. The data fed to the system is helpful for future predictions. In this way, the machine learning approach has been used in robotics, smart farming, the health industry, and many other sectors. Machine learning is basically subdivided into supervised, where the class label is known, and unsupervised, where the unknown class label is handled. The algorithms of machine learning can solve many security issues like spam detection, intrusion detection, and malware detection. A machine-learning-based subsystem has been proposed by many researchers to identify and mitigate an intrusion [7]. Firefly algorithm is used by authors in [8] for detecting and reducing attacks from intruders. The work presented a cyber IDS by combined feature selection algorithm. The authors in [12] address the issues and possibilities of improving network security through the use of ML. In 2021, a tree-based model for the purpose of intrusion detection was proposed by the authors [13]. The authors also considered some metrics in this paper, such as accuracy, recall, ROC values, and so on. The model's efficiency has been assessed using the values derived from experimental data. Some authors have categorized the intrusion detection system utilizing machine learning into two distinct groups: one that emphasizes anomalies and another that concentrates on signatures, taking into consideration byte sequencing or patterns of intrusion sequences. All these were achieved by using different algorithms, i.e. Decision Tree, KNN, SVM, and Random-Forest [14].

2.3. Why Machine Learning is so important for Intrusion Detection?

Machine learning is a globally recognized method for deriving insights from data without requiring explicitly written programs. Since it involves students discovering rules through examples, it is sometimes referred to as the inductive learning approach. Algorithms that can be trained to complete a task are the main goal of machine learning. It includes numerous techniques for in-depth analysis and efficient resolution of classification and regression issues. Because of its strength, this method can handle both labelled (supervised) and unlabeled (unsupervised) data. So, many good reasons exist for the use of machine learning for intrusion detection. [15].

3. Literature Reviews

Machine learning (ML) techniques are extensively used across multiple domains, such as disease prediction, weather forecasting, information retrieval, identity recognition, facial and voice recognition, and statistical processing. The recent application of ML in IDS has fostered urgency as a result of its potential to improve network infrastructure security. The incorporation of ML into IDS offers an adaptive framework that analyses network traffic patterns, enabling the detection of anomalous behaviours that may signify potential security threats. Researchers have investigated multiple machine learning models and techniques, such as decision trees, support vector machines, neural networks, and ensemble methods, to improve the efficacy of intrusion detection systems in identifying a range of dynamic cyberattacks. Ensemble methods, including the stacked ensemble approach, integrate multiple base models to enhance detection systems' accuracy, robustness, and generalisation, thereby providing a viable solution to the challenges presented by the complexity and variability of contemporary cyberattacks. The authors (Table 2) review recent work on this subject in this section.

In [16], the researchers found a way to find strange activities in networks that use CNN. The proposed model could detect abnormalities and gain impressive accuracy for the Network Intrusion Detection (NID) network. Researchers in [17] proposed a categorical cross-entropy coupled with the Adam optimization algorithm for the NID system. This resulted in a 94.4% accuracy for the testing dataset. Improved accuracy results for intrusion detection are achieved by authors in [18] by using an ML-based approach. The work in [19] delineates a hybrid convolutional recurrent neural network for NID. The simulation result achieved by the author shows an accuracy of 97.75% for the trained dataset. In [20], authors developed a DL-based approach and achieved an accuracy of over 90% for their proposed model. Researchers in [21] utilized a machine learning algorithm to attain an accuracy of 95.95% with the NSL-KDD dataset. The authors in [22] presented a novel and scalable DL-deep learning-based intrusion detection system. The evaluation of an Stacked Autoencoder SVM(SAE-SVM) scheme within a big data framework demonstrated an accuracy of 95.98%. Anthony et al.[56] works on intrusion detection systems for autonomous vehicles using Non-Tree based machine learning algorithms on the CICIDS2017, NSL-KDD, and CAN datasets, where NSL-KDD achieved an accuracy of 98.57%. Though it has higher accuracy, on the contrary, it resulted a smaller F1-Score, that is 98.79% on the NSL-KDD dataset.

The proposed model also required lower power and resource usage. Deep learning models like CNNs, LSTMs, and GRUs are used in this study [23]. The structure of the model is also enriched with a voting mechanism with an ensemble deep-learning architectural framework to enhance the computation and learning of hierarchical patterns. CNN-LSTM achieved an incredible 99.7% accuracy, and CNN-GRU achieved 99.6%. The F1 scores for the two models were also very high: 0.998 and 0.997.

An improved and efficient RCNN model is developed by the author to aggregate segmentation mask grading of mixed aggregates in [24]. The study was conducted through three distinct trials and indicated that AS Mask RCNN attained an impressive accuracy of over 89.13% across all testing conditions. It exceeded the performance of the faster RCNN and mask R-CNN models by 8.85%. Being quicker with an approximate 1.29 seconds processing time optimization, it fits better for the needs of near real-time filed identification in single image segmentation. In this paper [25], a new machine learning-based network intrusion detection that uses RO-Random Oversampling to solve the data imbalance. PCA–principal component analysis and stacking feature embedding are used with big and unbalanced datasets to reduce dimensions. Saba et al. [26] attained an accuracy of 99.51%; it is crucial to note that their methodology was based on a deep learning model, which generally requires larger datasets and comprehensive hyperparameter optimization. This frequently results in enhanced accuracy; however, it may also introduce greater computational complexity and diminish interpretability.

In [27], authors of this research proposed a deep learning-based stacked Nonsymmetrical Deep Auto-Encoder model for intrusion classification by using NSL-KDD as well as KDD Cup '99 datasets. The suggested model, relative to other prior works, increases efficiency by 5% and Minimis time by up to 98.81%. In [28], Authors utilize a deep learning approach centered on binary classification to differentiate between benign and malicious data packets, employing with NSL-KDD and CICIDS2017 datasets. The experimental findings indicate a notable accuracy and F-score for multiclass classification across both datasets.With the NSL-KDD dataset and many machine learning techniques used, the authors of [29] obtained an accuracy rate of 91.06%. This outcome shows a significant improvement over the performance of a single method. Using NSL-KDD and UNSW-NB15 datasets, [30] mostly seeks the most efficient intrusion detection machine learning approach and does a comparison study among six machine learning algorithms classified as supervised, semi-supervised, and unsupervised learning. None of the methods covered, nevertheless, efficiently solve the intrusion detection problem.

The research builds to augment the existing body of knowledge by improving efficacy of IDS-Intrusion Detection Systems. An improved overall performance of the IDS is achieved by the development of a binary classification system that differentiates between normal and problematic IoT data. This approach makes use of

Table 2. Summary Table of Related Works

Articles	**Methods**	**Major Findings**
Saba et al.[26]	Deep learning model	Attained accuracy is 99.51% and 92.85% respectively with NID and BoT-IoT datasets.
Rajadurai et al.[29]	Stacked model from weak classifiers	Achieved 91.06% accuracy, which is greater than ML algorithms like the Random Forest, ANN, and CNN
Sarhan et al.[18]	Machine learning model	Achieved improved results of accuracy and F1 score for multiple datasets
Song et al.[44]	CNN	Lesser False-negative rate comparing to the other conventional methods
Li et al.[36]	CNN fusion algorithms and DST-TL method	92.67% precision for probe attack and 68.82% accuracy for binary classification
Amouri et al.[42]	2-layer intrusion detection system considering Floods and Blackhole attacks	98% intrusion detection accuracy for higher node density which is 90% for lower node density
Latah et al.[40]	K-Nearest Neighbor and Neural Networks	91.2% accuracy was gained so far
Shone et al.[27]	Combination of Deep and Supervised Learning	Non-symmetric Deep Auto-Encoder model which minimizes training and testing time
Abdulsalam et al.[21]	Machine learning model for SDN	Achieved 95.95% accu-racy and classified the type of attack.
Mighan et al.[22]	Deep learning model	Achieved 95.98% accuracy for SAE-SVM scheme in big data framework
Muhammad et al. [19]	Hybrid Convolutional Recurrent Neural Network-Based NID System	Achieved an accuracy of 97.75% for datasets utilizing 10-fold cross-validation.
Anthony et al. [56]	Non-Tree Based Machine Learning Algorithms	Achieved 98.57% accuracy and 98.79% F1-Score on NSL- KDD dataset

a variety of ensemble classifiers and supervised machine-learning methods. This paper [32] presents a unique network IDS model based on an Ensemble Learning algorithm. The significant feature selection is achieved through an approach that combines CFS–FPA: Correlation Feature Selection with Forest Panelized Attributes. The enhanced intrusion detection used ensemble learning algorithms such as AdaBoosting and bagging to refine four classifiers: K-Nearest Neighbor, Support Vector Machine, RandomForest, and NaiveBayes. The offered system [33] employs pre-processing methods to accomplish classification efficiency and injects different ML-machine learning algorithms. The performance is improved by using this ensemble technique called stacking, which combines three different base models RF-Random Forest, Decision Tree, and k-Nearest Neighbors and one meta-model Logistic Regression. The experimental results with UNSW-NB15 dataset showed that the accuracy of proposed IDS in training and testing stages were 96.16% and 97.95% while their precision rates were 97.78% and 98.40%, respectively [26]. This research [34] examines the performance of CADS across three feature selection methods: RFE-Recursive Feature Elimination, MI-Mutual Information and LFS-Lasso Feature Selection. It proposes a new stacked ensemble classification method which incorporates with Random Forest, XGBoost, and Extra-Trees classifiers alongside a Logistic Regression meta-model. The research in [35] presents Tachyon, which incorporates multiple statistical and tree-based AI-Artificial Intelligence techniques, including XGBoost-Extreme Gradient Boosting , RF-Random Forest, BART-Bidirectional Auto-Regressive Transformers , LR-Logistic Regression, MARS-Multivariate Adaptive Regression Splines, DT-Decision Tree, and a top k stack ensemble, to differentiate between normal and malicious attacks in a binary classification context. IoTID2020 dataset, comprises 625,783 samples and includes 83 features. The stacked ensemble demonstrated performance of 99.8%, surpassing the baseline approaches. While significant progress has been made in detecting malicious activity within the networking system, a concerning aspect remains in the shape of the duration required to formulate an adequate response.

Furthermore, there are two different categories of classification existing in this area of research. These are termed binary and multiclass classification. Authors in [22], modeled a unique method based on support vector machine as the classifier. The generated results with an accuracy of 95.98% when compared with other models resulted for a smooth response time and efficient NIDS. The study showed notable improvement in accuracy for multiclassification as well as binary classification. The authors introduced a novel E-graphSAGE-based intrusion detection method for Internet of Things systems. However, this work presented did not address the issue regarding runtime, in contrast to the work demonstrated in [22]. Multiple studies are currently underway aimed at enhancing intrusion detection capabilities. In certain studies, as referenced in [36], the authors conducted a comparison and a concise analysis of intrusion detection within networking systems utilizing machine learning techniques. The authors in [37] presented an approach based on recurrent neural networks utilizing deep learning techniques. The survey results may provide valuable insights for researchers addressing challenges within this field.

In summary, the analysis indicates that a significant portion of research regarding intrusion detection utilizing machine learning algorithms remains in the developmental stage. Most of the study is survey-based and the accuracy is 96% on average. That is why, the authors got motivated to work on detecting an intruder through the powerful Stacked model of ML.

4. Proposed Methodology of ML-based Models

In recent days, several machine learning models have been utilized extensively for anomaly-based intrusion detection, including Deep Neural Networks (DNN), Random Forest (RF), Stochastic Gradient Descent (SGD), and Logistic Regression (LR). To detect intrusion, in [38] the authors propose ML-based approaches in the SDN environment efficiently. They develop a Stacked technique based on the existing techniques of ML, as shown in Fig. 1. Moreover, this work highlights the need to improve the networking model's accuracy. The authors in [39] achieved an accuracy of 98% based on the experimental data sets. For enhancing model performance, there are numerous ensemble methods like bagging, boosting, and stacking which are employed for intrusion-based anomaly detection systems [57]. Building on these established techniques, the authors of this paper present a novel stacked ensemble method that combines LR, RF, SGD, and DNN into a single framework. The innovative aspect of this work is the construction of a heterogeneous stacking model designed for intrusion detection, in which a meta-learner optimally combines the advantages of various base models. By utilizing the complementary strengths of both deep learning models and classical machine learning methods, our model achieves a more thorough pattern detection than previous ensemble approaches, which are described briefly one by one in the subsidiary section.

4.1. Dataset

The NSL-KDD dataset is preferred in this research, which is an improved version of the KDD-Cup-99 dataset with many erroneous entries [40]. The researchers in this work have historically used the NSL-KDD dataset to assess public presentation of NIDS-Network Intrusion Detection Systems. Consequently, there are several notable assessment outcomes available for comparison. Furthermore, in [41] the NSL-KDD dataset contains 39 attacks, where an individual attack is categorized as U2R, DoS, R2L, and Probe. This dataset has 125,973 examples of network activity that can be used for training and 22,554 examples that can be used for testing. In addition, each sample has 41 traits which are separated into three groups: traffic-based features, content-based features, and basic features . Detection rates exceed 98% in high power or node velocity scenarios, whereas they decrease to approximately 90% for lower data conditions. Power and node velocity scenarios [42]. Furthermore, the experiment

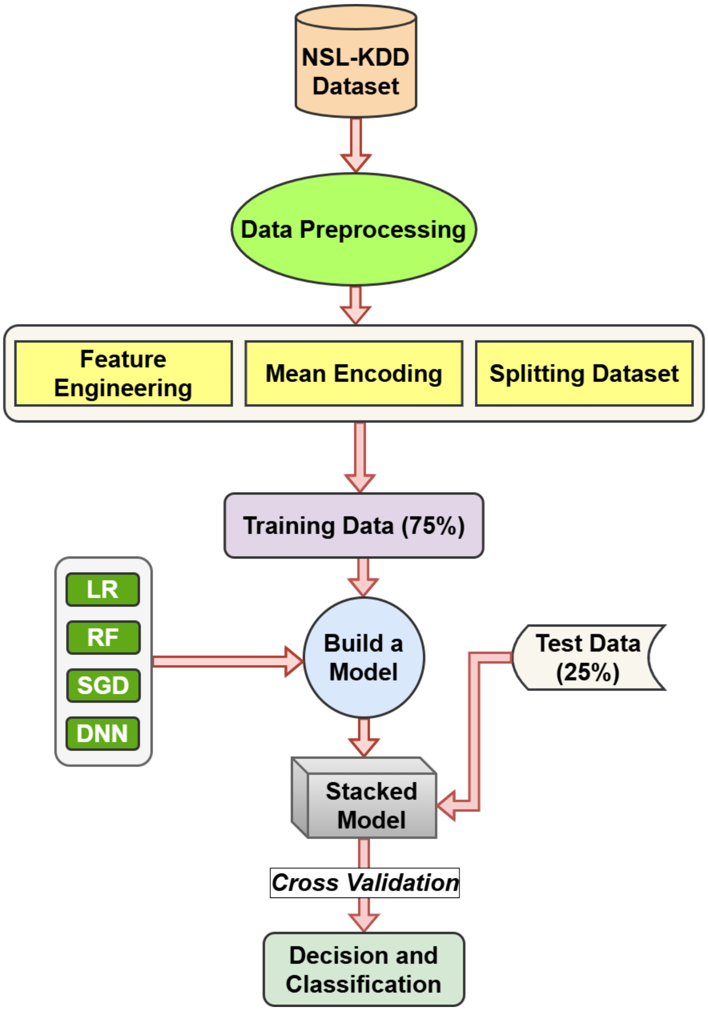

Figure 1. Proposed Architecture for Stacked Model

conducted in this study involved selecting all the features before dividing data points in training and testing purposes. The SDN environment is capable of generating such data with ease.

4.2. Data Preprocessing

The authors used Feature Engineering and Mean Encoding techniques in the data preprocessing part.

4.2.1. Feature Engineering The feature engineering stage in machine learning is crucial for the conversion of unprocessed data into usable features, which are subsequently applied in accordance with equations 1, 2, and 3 to develop a prediction model through machine learning or statistical modeling techniques.

$$W = \sum_{j=1}^{M} \sum_{k=1}^{n_j} w_{jk} x_{jk} \tag{1}$$

$$T = \sum_{j=1}^{M} \sum_{k=1}^{n_j} t_{jk} x_{jk} \tag{2}$$

$$R = \sum_{j=1}^{M} \sum_{k=1}^{n_j} r_{jk} x_{jk} \tag{3}$$

In 1 W is the weight factor that sums up all of the data elements called X for each column M and each row n. Similarly in 2, T is the time factor, and in 3, R is the rate of change required for each of the data elements. The goal of feature engineering in ma-chine learning is to improve the model's performance.

4.2.2. Mean Encoding Similar to label encoding, mean encoding differs in that the target and labels are correlated. For instance, in mean target encoding, the mean value of the target variable on a training dataset is used to make encoding decisions for each category in the feature label. Subsequently, the NSL-KDD dataset is divided into two segments, allocating 75% data in training and reserving 25% in testing purposes. Subsequently, a set of machine learning algorithms is utilized on the training data to develop the optimal model. This study, as outlined in Section 4.3, utilizes a set of machine learning algorithms.

4.3. Machine Learning Approaches

In cyber and network security problems, ML approaches are used widely. The authors employed a bunch of ML algorithms to create the best model in this research, including SGD-Stochastic Gradient Descent, RF-Random Forest, DNN(Deep Neural Network), and Stacked Model[43]. Now, the models are discussed all below:

4.3.1. Logistic Regression (LR) A ML based algorithm based on a statistical concept is actually an S-shaped graph given in equation 4 below. It can take any real number as input and it then maps them to values in the range [0,1]. A Binary LR has been used in this research work.

$$y = e^{\wedge(c_0+c_1x)} / \left(1 + e^{\wedge(c_0+c_1x)}\right) \tag{4}$$

y is the predicted value/output, c_0 is the intercept and c_1 is the co-efficient/weight. Authors train a separate LR model for each attack type. Then, the arithmetic mean of the accuracy of all models is calculated.

4.3.2. Random Forest (RF) A very popular classification algorithm in Ma-chine Learning is RF-random Forest. It often out-performs other classification algorithms. Actually, a bunch of decision trees are generated in Random Forest. In real terms, it is an improved version of bagged decision trees. The paper [44] presents a DCNN-deep convolutional neural network based IDS-intrusion detection system that is designed to protect the CAN bus of the vehicle. In Fig. 2, Binary RF models (for each attack type) are used in this investigation. The average of probabilistic prediction of RF models is calculated rather than voting for a single class proposed in [42].

4.3.3. Stochastic Gradient Descent (SGD) SGD, being actually an optimized algorithm, searches for the parameter of other algorithms for which the cost function is minimized using the random variable. For high dimensional dataset in paper [45] authors proposed a model for intrusion detection. It starts with initial values for parameters and then searches for expected values as shown in Fig. 3. In this investigation, authors use a binary linear Support Vector Machine equation 5 to use in SGD.

$$lossFunction(y_i, f(x_i)) = \max(0, 1 - y_i f(x_i)) \tag{5}$$

4.3.4. Deep Neural Network (DNN) A neural network operates based on the principles of human brain function. Figure 4. The human brain is composed of neurons. The neurons transmit electrical signals between them when stimulated. In [46], authors introduced a DDoS attack mitigation scheme for ISP networks utilizing ML in conjunction with SDN. This study focuses on detecting DDoS attacks in large-scale SDN-based cloud environments. A neuron, within the framework of a neural network, aggregates the outputs from its connected neurons and activates if the resulting value exceeds a specified threshold [47]. The sum, in this case, is a

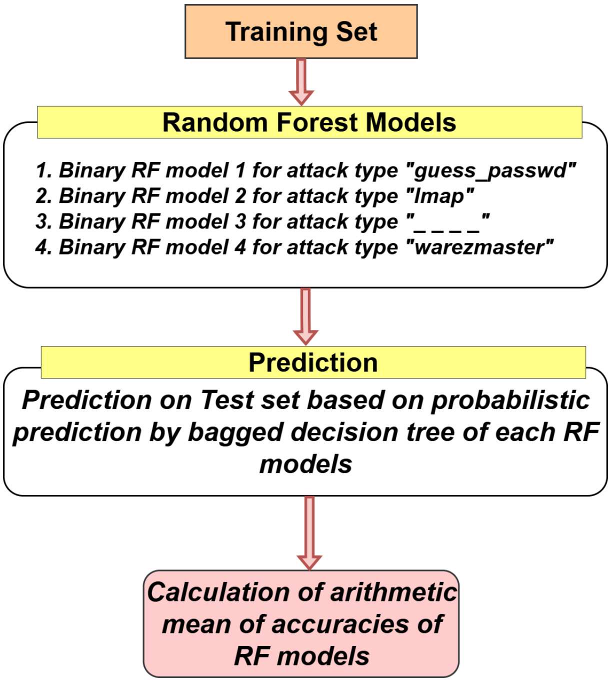
Training Set
Random Forest Models
1. Binary RF model 1 for attack type "guess_passwd"
2. Binary RF model 2 for attack type "Imap"
3. Binary RF model 3 for attack type "_ _ _ _"
4. Binary RF model 4 for attack type "warezmaster"
Prediction
Prediction on Test set based on probabilistic prediction by bagged decision tree of each RF models
Calculation of arithmetic mean of accuracies of RF models

Figure 2. Random Forest (Bagged Decision Tree) on NSL-KDD dataset

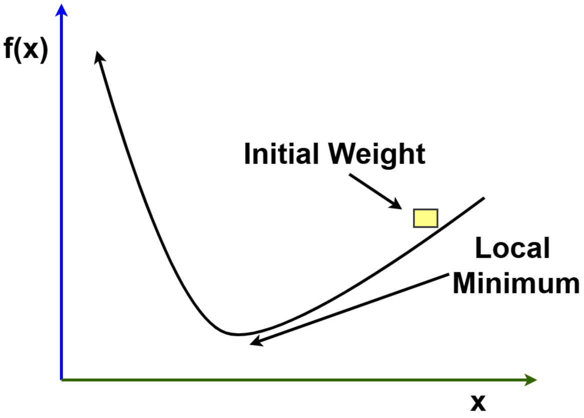

Figure 3. Concept of Gradient Descent

weighted sum. The weights are the learnable parameters. A final output layer usually has one input layer and numerous hidden levels. A neural network is classified as a deep neural network when it contains multiple hidden layers. Figure 4 illustrates the overall architecture of a deep neural network. The algorithms demonstrate notable performance improvements, even when handling extensive datasets. This research employs the Rectified Linear activation function (ReLU) in the first, third, fifth, and sixth layers of a seven-layer binary DNN, with the seventh layer utilizing the sigmoid (logistic) activation function and normalization has been implemented in the second and fourth layers as described in Table 3. The required hyperparameters like optimizer, loss function, number of epochs, batch size, matrices and their corresponding values are provided in Table 4.

4.3.5. Stacked Model This study uses an ensemble (stacking) of the models (LR+RF+SGD+DNN) to achieve better accuracy than the accuracy obtained by individual models. Incorporating numerous ML models into just one

Table 3. DNN Architecture for Binary Classification

Layer Type	**Layer Details**	**Activation Function**	**Units**
Input Layer	Input dimension (features)	-	41
Dense Layer 1	Fully connected layer	ReLU	1024
Batch Normalization	Normalizes activations for layer 1	-	-
Dense Layer 2	Fully connected layer	ReLU	1024
Batch Normalization	Normalizes activations for layer 2	-	-
Dense Layer 3	Fully connected layer	ReLU	512
Dense Layer 4	Fully connected layer	ReLU	64
Dense Layer 5 (Output)	Fully connected layer (final output)	Sigmoid	1

Table 4. Hyperparameter of DNN Architecture

Hyperparameter	**Value/Details**
Optimizer	Adam (Adaptive moment estimation)
Loss Function	Binary Cross-Entropy
Activation Functions	ReLU (Hidden layers), Sigmoid (Output layer)
Batch Normalization	Yes, applied after each ReLU activation
Number of Epochs	50
Batch Size	1024
Metrics	Accuracy

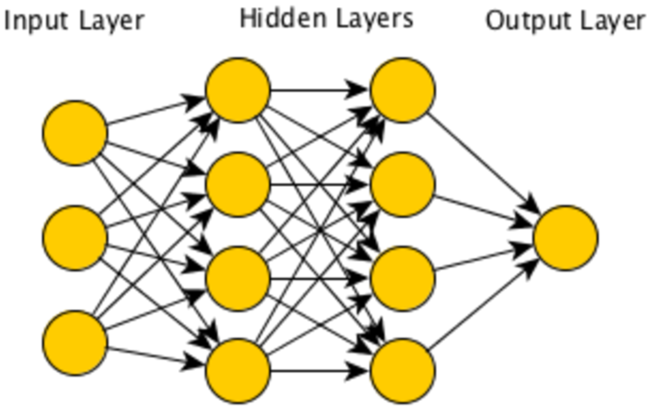

Figure 4. Deep Neural Network

framework is known as an ensemble approach. It is used when researchers lack confidence in a certain forecasting model. To generate more reasoning than a single model, a collection of weak models is used; it is a widely used method for combining the advantages of several models [48]. Figure 5 presents the fundamental stack approach. There are two kinds of model ensembles. The first model is the homogeneous collective model, while the second is the heterogeneous ensemble model. The homogeneous ensemble model utilizes a single type of classifier, unlike the heterogeneous ensemble model, which incorporates various classifiers. Homogeneous ensemble models consist of techniques such as bagging and boosting, while heterogeneous ensemble models include methods like stacking. The various machine learning models in stacking enhance predictions by leveraging the strengths of different models [49], resulting in improved reasoning. Stacking is commonly known as blending. This approach involves a stacking generalization that considers various weak or simple learners and trains them simultaneously.

Here is a more detailed breakdown of the stacking process:

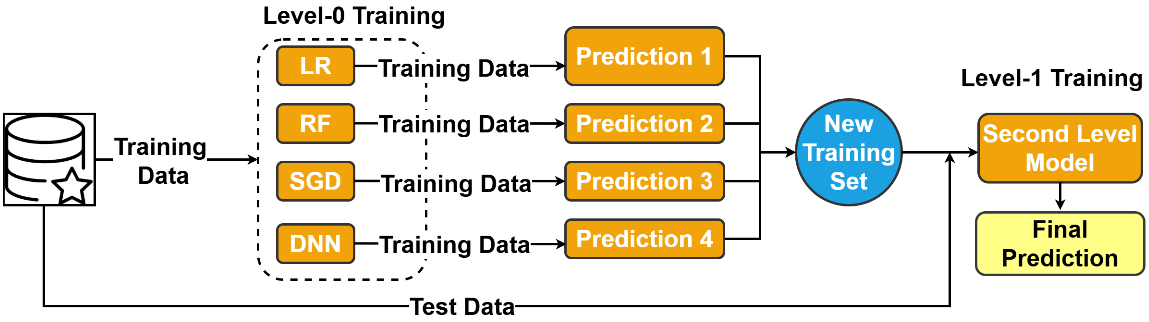

Figure 5. Stacked Method

- **Training the Base Models:** The base models are trained independently using the training dataset. The models exhibit variations in architecture and learning algorithms, offering distinct perspectives on the data. Utilising various models enables the ensemble to encompass a broad spectrum of patterns within the data.
- **Generating Predictions:** Upon completion of training the base models, they are employed to generate predictions on the training dataset. These forecasts, termed level-1 predictions, are subsequently aggregated. The predictions from the base models may consist of either class labels or probabilities, contingent upon the specific issue configuration.
- **Combining Predictions:** The subsequent stage is to incorporate these forecasts into a conclusive forecast. This is accomplished with an alternative model, the meta-model or stacking model. In our instance, we employed a straightforward classifier, such as Logistic Regression (LR), or an alternative model that optimally integrates the predictions from the foundational models. The meta-model is trained with level-1 predictions as input and the actual labels as output. The average stacking method combines the predictions from the level 1 models, which can be used as given in eqn. 6. The outcomes of the weak learners or the level 1 learners need to be combined into a final prediction.
- **Final Predictions:** Upon training the meta-model, it produces the final prediction derived from the outputs of the basis models. This method's efficacy stems from the meta-model's ability to optimally integrate the predictions of base models, hence enhancing overall performance by addressing the shortcomings of individual models.

The authors created a stacked model of four weak learners. These models include LR-Logistic Regression, RF-Random Forest, SGD-Stochastic Gradient Descent, and DNN-deep learning neural networks [50]. These models use different measures to predict the outcome and run in parallel. They have their strengths and weaknesses. These are our level 1 models.

$$\hat{y}_{\text{final}} = \frac{1}{N}\sum_{i=1}^{N}\hat{y}_i \tag{6}$$

Algorithm 1 presents the pseudocode for the stacked ensemble model.

4.3.6. Performance Metrics The evaluation of all models is conducted through the confusion matrix, which comprises four values: TP-True Positive, TN-True Negative, FP-False Positive, and FN-False Negative, as illustrated in Table 5. Overall, NIDS output is assessed using different terms of performance metrics. They could be mentioned as the rate of predicting true positives, which is known as precision (P), the overall accuracy (AC), F-score, recall (R), and False Positive Rate (FPR). A matrix called confu-sion matrix, consisting of the value of TP-True Positive is the data points that were tested as attacks with equal ground values. TN-True Negative is a proper description of the number of non-attack records. FP-False Positives refer to records that are incorrectly identified

Algorithm 1 Stacked Ensemble Model for Intrusion Detection

```
Input: Preprocessed dataset $\mathscr{D}$ with features $\mathbf{X}$ and labels $\mathbf{y}$
Output: Predicted labels $\hat{\mathbf{y}}$ for test data
Step 1: Preprocessing
Feature Engineering and Mean Encoding.
Step 2: Train Base Models
for model $M \in$ {Logistic Regression, Random Forest, SGD, Deep Neural Network} do
    Train $M$ on 75% of the training data.
    Generate predictions $\hat{\mathbf{y}}_M$.
end for
Step 3: Train Meta-Model
Combine predictions from base models as meta-features.
Use Logistic regression as meta model.
Train a stacked classifier $M_{\text{meta}}$ using cross-validation.
Step 4: Prediction and Evaluation
Predict final labels $\hat{\mathbf{y}}$ using $M_{\text{meta}}$ on test data.
Evaluate using metrics: Accuracy, Precision, Recall, and F1 score.
```

as normal, whereas FN-False Negatives indicate the number of attack records that have been misclassified. The following formulas are used to compute these metrics:

$$AC = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{7}$$

$$P = \frac{TP}{(TP+FP)} \tag{8}$$

$$R = \frac{TP}{(TP+FN)} \tag{9}$$

$$FPR = \frac{FP}{(TN+FP)} \tag{10}$$

$$FNR = \frac{FN}{(TP+FN)} \tag{11}$$

$$F1-Score = 2\times\frac{\mathrm{P}\times\ \mathrm{R}}{(\mathrm{P}+\mathrm{R})} \tag{12}$$

Table 5. Confusion Matrix

Predicted	**Normal**	**Attack**
Actual Normal	True Negative (TN)	False Positive (FP)
Actual Attack	False Negative (FN)	True Positive (TP)

5. Experimental Result (Case Study) and Discussion

Based on the effectiveness of various classifier mod-els, including Logistic Regression, Stochastic Gradi-ent Descent, Random Forest Classifier, Deep Neural Network, and Stacked Model, a thorough analysis was carried out. Metrics including accuracy, recall, precision, F-score, and false positive rate are highlighted in the Table 6, which shows the model's performance on the test dataset. Here is a synopsis of the results shown in Table 6:

Table 6. Overall performance of ML models for NSL-KDD Dataset

Model	Fold	Accuracy	Precision	Recall	F1-score
Logistic Regression	1st	98.9%	99.6%	97.9%	98.7%
	2nd	98.9%	99.5%	98.2%	98.99%
	3rd	98.9%	99.3%	98.4%	99.1%
	4th	98.8%	99.2%	98.1%	98.8%
	5th	98.8%	99.4%	98.2%	98.9%
	Mean	**98.9%**	**99.5%**	**98.2%**	**98.8%**
Stochastic Gradient Descent	1st	93.6%	98.1%	94.2%	96.2%
	2nd	93.7%	97.9%	94.5%	96.1%
	3rd	94.1%	98.0%	94.3%	96.0%
	4th	93.9%	98.1%	94.2%	96.3%
	5th	94.4%	98.0%	94.3%	96.2%
	Mean	**93.9%**	**98.0%**	**94.3%**	**96.1%**
Random Forest Classifier	1st	99.9%	99.9%	98.6%	99.0%
	2nd	99.9%	99.7%	98.5%	98.8%
	3rd	99.8%	99.8%	98.7%	99.3%
	4th	99.9%	99.6%	98.2%	99.2%
	5th	99.9%	99.8%	98.5%	99.2%
	Mean	**99.9%**	**99.8%**	**98.5%**	**99.1%**
Deep Neural Network	1st	99.2%	99.8%	98.6%	98.9%
	2nd	99.2%	99.5%	98.5%	99.0%
	3rd	98.4%	99.6%	98.6%	98.7%
	4th	99.3%	99.8%	98.5%	98.8%
	5th	99.1%	99.9%	98.7%	98.4%
	Mean	**99.1%**	**99.7%**	**98.6%**	**98.9%**
Stacked Model	1st	99.98%	99.9%	98.5%	99.0%
	2nd	99.9%	99.8%	98.8%	99.0%
	3rd	99.9%	99.6%	98.6%	99.6%
	4th	99.9%	99.8%	98.4%	99.2%
	5th	99.9%	99.9%	98.8%	98.4%
	Mean	**99.9%**	**99.8%**	**98.6%**	**99.2%**

- **Accuracy:** Stacked Classifier has the highest test accuracy of all models, which is (99.9%). Authors compare the accuracy of other classifiers like Logistic Regression (98.9%), Random Forest (98.3%), Stochastic Gradient Descent (98.0%), and Deep Neural Network (99.1%).
- **Recall:** Stacked Model and Deep Neural Network (98.6%) has the highest recall score of 98.6, which means it has correctly identified all non-attacks. While the recall of other classifiers like Logistic Regression (98.2%), Random Forest (98.5%), Stochastic Gradient Descent (94.3%).
- **Precision:** Stacked Model and Random Forest Classifi-er have a 99.8 maximum level of precision. While the precision of other classifiers like Logistic Regression (99.5%), Stochastic Gradient Descent (98%), and Deep Neural Network (99.7%) accuracy.
- **F1-score:** For the Stacked model with an accuracy of 99.9, the best F1-score is (99.2%). While the F1-score of other classifiers like Logistic Regression (98.8%), Random Forest (99.1%), Stochastic Gradient Descent (96.1%), and Deep Neural Network (98.9%).

The accuracy of the proposed model is 99.9% whereas SVG, RFC, DNN, and LR are 93.9%, 99%, 99.1%, and 98.9% respectively. Now the accuracy of all the models is represented by a bar chart below in Fig. 6 as well as in tabular form (Table. 8).

To evaluate how well different theories work, one might use a confusion matrix. It is possible to see the model's TP, TN, FPN, and FN values in the confusion matrix. The suggested stacking model is included in the confusion matrix for all models, as shown in Table 7.

Table 7. Confusion Matrix Components (TP, FP, TN, FN) for Different Models.

Model	**TP**	**TN**	**FP**	**FN**
	(Attack Predicted as Attack)	**(Normal Predicted as Normal)**	**(Normal Predicted as Attack)**	**(Attack Predicted as Normal)**
LR	22175	329	12	45
SGD	20800	420	40	20
RF	22180	329	36	40
DNN	22011	417	15	72
Stacked Model	22190	320	15	40

The stacked model demonstrated a more significant reduction in FP-false positive and FN-false negative values than the other models, as indicated by the confusion matrix. The findings suggest that the stacked model outperforms other models.

False Positive Rate (FPR)

The Stacked model has False Positive Rate (0.0448). While the FNR of other classifiers such as Logistic Regression (0.0352), Stochastic Gradient Descent (0.0870), Random Forest (.0986), and Deep Neural Network (0.0347).

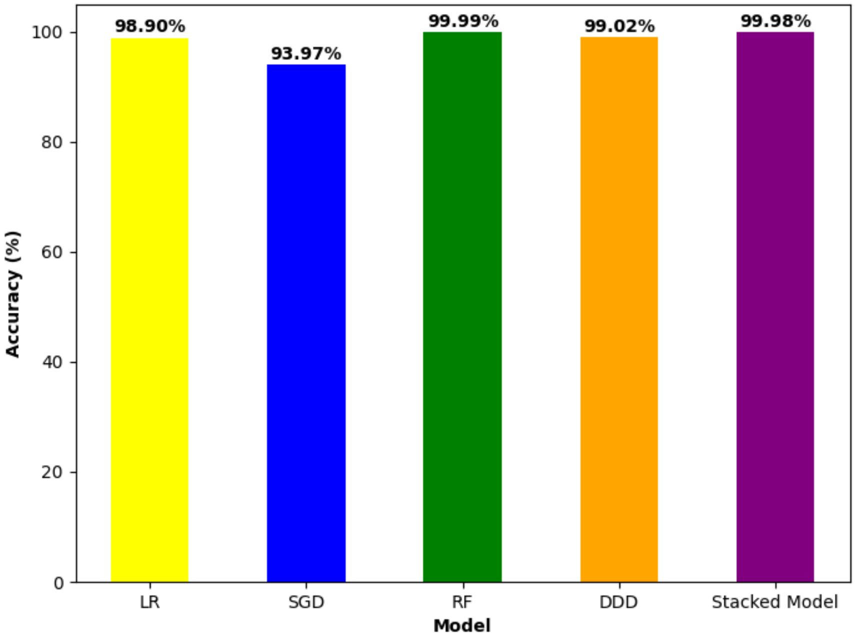

Figure 6. Experimental Models Comparison

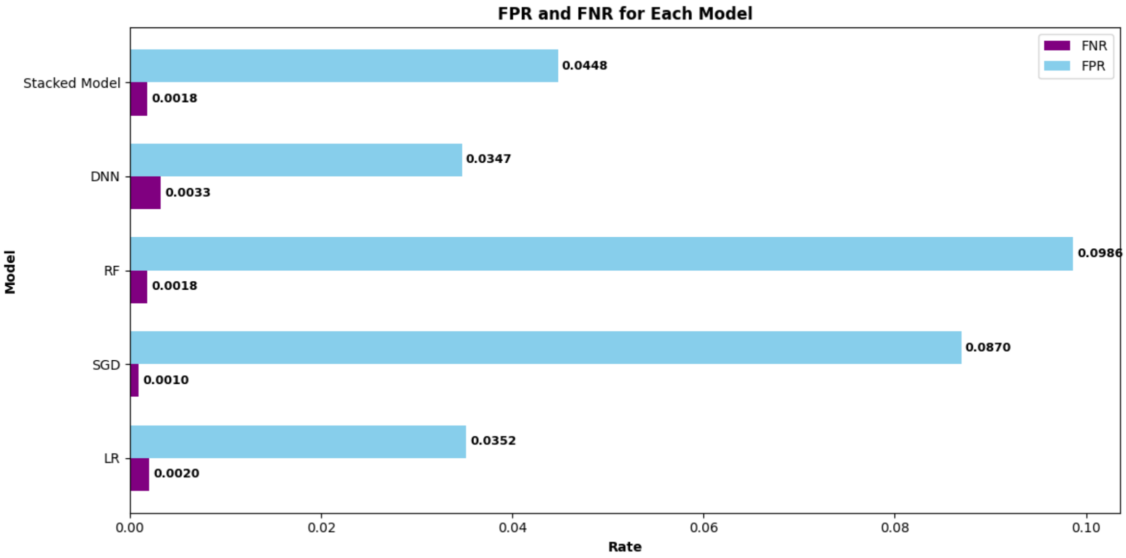

Figure 7. FPR and FNR of all ML Model

False Negative Rate (FNR)

The Stacked model demonstrates the lowest False Negative Rate at 0.0018. The false negative rates of various classifiers include Logistic Regression at 0.0020, Stochastic Gradient Descent at 0.0010, Random Forest at 0.0018, and Deep Neural Network at 0.0033. The bar chart below illustrates the FPR-false positive rate and FNR-false negative rate for all models in Fig. 7. The stacked model is currently undergoing a comparison with existing models. Table 8 present a comparison of the stacked model with other established research methods.

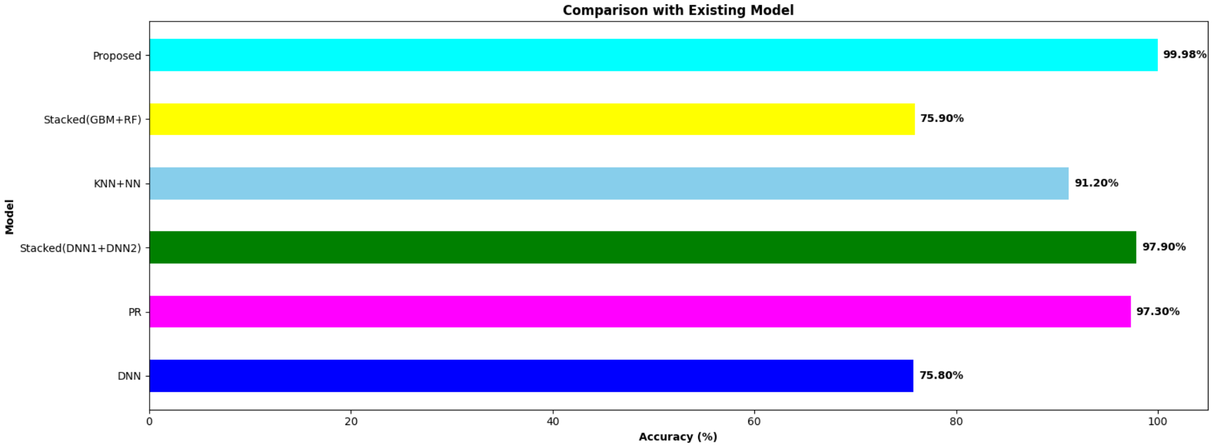

Figure 8. Comparison of the Proposed Model with Existing Model

The comparison of accuracy between the leading existing models from various authors and the proposed stacked model is illustrated in Fig. 8. The efficiency of the Stacked model is evaluated in comparison to other research methods presented in Table 8.

A pattern recognition method utilized by Abubakar et al. demonstrated an accuracy exceeding 97 percent [51]. Samriddhi et al. applied the Naive Bayes technique [52], while Nisharani et al. utilized support vector classification,

Table 8. Comparison of Deep Learning-Based IDS Models

Study	**Method**	**Dataset**	**Accuracy (%)**	**F1-Score**	**Advantages**	**Limitations**
Tuan et al. [53]	DNN	NSL-KDD	75.8	0.75	Effective feature learning	Lower accuracy, overfitting risk
Atiku et al. [51]	Pattern Recognition	NSL-KDD	97.3	0.97	Good performance on structured data	Limited scalability for large datasets
Shone et al. [27]	Stacked (DNN1 + DNN2)	NSL-KDD	97.9	0.98	Improved feature representation	High computational cost
Latah et al. [40]	KNN + NN	NSL-KDD	91.2	0.91	Hybrid model improves detection	Inefficient for real-time scenarios
Rajadurai et al. [29]	Stacked (GBM+RF)	NSL-KDD	75.9	0.75	Robust against imbalanced data	Lower accuracy than deep learning models
Proposed SIDS Model	**Stacked (LR + RF + SGD + DNN)**	**NSL-KDD**	**99.98**	**0.99**	**High accuracy, robust ensemble learning**	**Requires fine-tuning for different datasets**

with both methods demonstrating an accuracy exceeding 80%. Ultimately, Tang et al. demonstrated that the DNN approach can reach accuracy levels exceeding 75% [53]. Latah et al. demonstrated accuracy exceeding 90% through the integration of K Nearest Neighbor and Neural Network [40]. On the same dataset, the offered Stacked model demonstrated an accuracy exceeding 98 percent. Figure 7 presents a comparison between the models and the offered model. The study [54] compiled three distinct methods in machine learning to provide a validated prediction output utilizing a comprehensive dataset and confirmed their applicability. The ensemble machine learning method XGBoost is employed in paper [55] to achieve the anticipated outcome with optimal accuracy.

6. Limitations and Future Works

This paper uses the NSL-KDD dataset, which is known to have certain limitations, such as being outdated and not reflecting modern attack patterns. To overcome this issue in the future, we will apply GAN to generate synthetic data and combine it with CIC-IDS2017 and CIC-IDS2018, which includes modern attack types such as DDoS, botnets, web attacks, and ransomware. Furthermore, it will use blockchain technology to mitigate security issues and will verify methods with multiple heterogeneous datasets. The authors will execute the proposed method for large-scale data architecture. The integration of blockchain with an SDN will be explored to increase security and privacy within the intrusion detection domain. Additionally, the integration of big data, blockchain, and SDN-Software Defined Networking to improve security and privacy presents a promising strategy. A substantial amount of data will be utilized, accompanied by real-time analytics and dynamic network segmentation.

7. Conclusion

This research offers a unique way to intrusion detection called the SIDS-Stacked Intrusion Detection System, which employs ML-machine learning techniques. The authors employ popular ML-machine learning algorithms such as LR-Logistic Regression, RF-Random Forest, SGD-Stochastic Gradient Descent, and DNN-Deep Neural Networks. The suggested model achieves an excellent accuracy of 99.9% by employing the upcoming ”Stacked” ML methodology, outperforming previous techniques. Future projects will result in a comprehensive framework for strengthening network security, including sophisticated intrusion detection models. The study will look at the performance and scalability of SDN-software defined networking setups that use advanced DL-deep learning methods such as CNNs-Convolutional Neural Networks and RNNs-Recurrent Neural Networks for load balancing.

Acknowledgement:

Much obliged to everyone who helped with the research for this article.

Authors Contributions:

Each author had an equal hand in writing this piece. The article was also painstakingly vetted by all of the researchers.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process:

The authors used Grammarly and ChatGPT to make the writing more clear and easy to read when they were putting this piece together. The authors fully accepted responsibility for the publication's content after utilizing this tool or service, carefully going over it and making any necessary changes.

Declaration of Competing Interest:

The researchers affirm that they are not involved in any conflicts of interest.

Data Availability:

The source code is available via the GitHub link. The proposed approach and schematic diagram are shared with the scientific community for public access: https://github.com/jarin188/NSL-KDD.

REFERENCES

1. M. M. Nair, A. Deshmukh, and A. K. Tyagi, "Artificial intelligence for cyber security: Current trends and future challenges," in *Automated Secure Computing for Next-Generation Systems*, Wiley Online Library, 2024, pp. 83–114.
2. A. H. Janabi, T. Kanakis, and M. Johnson, "Survey: Intrusion Detection System in Software-Defined Networking," *IEEE Access*, 2024.
3. A. Rahman, M. J. Islam, A. Montieri, M. K. Nasir, M. M. Reza, S. S. Band, A. PescapÃ¨, M. Hasan, M. Sookhak, and A. Mosavi, "Smartblock-sdn: An optimized blockchain-sdn framework for resource management in iot," *IEEE Access*, vol. 9, pp. 283 61–283 76, 2021.
4. A. Rahman, M. K. Nasir, Z. Rahman, A. Mosavi, S. Shahab, and B. Minaei-Bidgoli, "Distblockbuilding: A distributed blockchain-based sdn-iot network for smart building management," *IEEE Access*, vol. 8, pp. 140 008–140 018, 2020.
5. Pabon Shaha, Md Saikat Islam Khan, Anichur Rahman, Mohammad Minoar Hossain, Golam Mahamood Mammun, and Mostofa Kamal Nasir, "A Prevalent Model-based on Machine Learning for Identifying DRDoS Attacks through Features Optimization Technique," *Statistics, Optimization & Information Computing*, 2024.
6. A. Rahman, J. Islam, D. Kundu, R. Karim, Z. Rahman, S. S. Band, M. Sookhak, P. Tiwari, and N. Kumar, "Impacts of blockchain in software-defined internet of things ecosystem with network function virtualization for smart applications: Present perspectives and future directions," *International Journal of Communication Systems*, p. e5429, 2023.
7. Islam N, Shamim SM, Rabbi MF, Khan MS, Yousuf MA. Building Machine Learning Based Firewall on Spanning Tree Protocol over Software Defined Networking. InProceedings of International Conference on Trends in Computational and Cognitive Engineering 2021 (pp. 557-568). Springer, Singapore.
8. Selvakumar, B.; Muneeswaran, K. Firefly algorithm based feature selection for network intrusion detection. *Computers & Security* **2019**, *81*, 148–155.
9. Md. Jahidul Islam, Anichur Rahman, Sumaiya Kabir, Md. Razaul Karim, Uzzal Kumar Acharjee, Mostofa Kamal Nasir, Shahab S. Band, Mehdi Sookhak, and Shaoen Wu. Blockchain-sdn-based energy-aware and distributed secure architecture for iot in smart cities. *IEEE Internet of Things Journal*, 9(5):3850–3864, 2022.
10. Manso, P.; Moura, J.; Serrão, C. SDN-based intrusion detection system for early detection and mitigation of DDoS attacks. *Information* **2019**, *10*, 106.
11. Swami, R.; Dave, M.; Ranga, V. Voting-based intrusion detection framework for securing software-defined networks. *Concurrency and Computation: Practice and Experience* **2020**, p. e5927.
12. Amrollahi Biouki, M.; Hadayeghparast, S.; Karimipour, H.; Derakhshan, F.; Srivastava, G. Enhancing network security via machine learning: opportunities and challenges. In *Advances in Intelligent Systems and Computing*; Springer: Cham, **2020**. https://doi.org/10.1007/978-3-030-38557-6_8.
13. Al-Omari, M.; Rawashdeh, M.; Qutaishat, F.; Alshira'H, M.; Ababneh, N. An Intelligent Tree-Based Intrusion Detection Model for Cyber Security. *Journal of Network and Systems Management* **2021**, *29*. https://doi.org/10.1007/s10922-021-09591-y.
14. Md S. H. Rabbi, Md M. Bari, T. Debnath, A. Rahman, A. K. Das, M. P. Hossain, and G. Muhammad, "Performance evaluation of optimal ensemble learning approaches with PCA and LDA-based feature extraction for heart disease prediction," *Biomedical Signal Processing and Control*, vol. 101, p. 107138, 2025.

15. A. Rahman, T. Debnath, D. Kundu, M. S. I. Khan, A. A. Aishi, S. Sazzad, M. Sayduzzaman, and S. S. Band, "Machine learning and deep learning-based approach in smart healthcare: Recent advances, applications, challenges and opportunities," *AIMS Public Health*, vol. 11, no. 1, pp. 58–109, 2024.
16. Morozova, O.; Nicheporuk, A.; Tetskyi, A.; Tkachov, V. Methods and technologies for ensuring cybersecurity of industrial and web-oriented systems and networks. *RADIOELECTRONIC AND COMPUTER SYSTEMS* **2021**, 145–156. https://doi.org/10.32620/reks.2021.4.12.
17. Ashiku, L.; Dagli, C.H. Network Intrusion Detection System using Deep Learning. *Procedia Computer Science* **2021**, *185*, 239–247. https://doi.org/10.1016/j.procs.2021.05.025.
18. Sarhan, M.; Layeghy, S.; Portmann, M. Towards a Standard Feature Set for Network Intrusion Detection System Datasets. *Mobile Networks and Applications* **2022**, *27*, 1–14. https://doi.org/10.1007/s11036-021-01843-0.
19. Khan, M.A. HCRNNIDS: Hybrid Convolutional Recurrent Neural Network-Based Network Intrusion Detection System. *Processes* **2021**, *9*, 834. https://doi.org/10.3390/pr9050834.
20. Rajesh Kanna, P.; Santhi, P. Unified Deep Learning approach for Efficient Intrusion Detection System using Integrated Spatial–Temporal Features. *Knowledge-Based Systems* **2021**, *226*, 107132. https://doi.org/10.1016/j.knosys.2021.107132.
21. Alzahrani, A.; Alenazi, M. Designing a Network Intrusion Detection System Based on Machine Learning for Software Defined Networks. *Future Internet* **2021**, *13*, 111. https://doi.org/10.3390/fi13050111.
22. Mighan, S.; Kahani, M. A novel scalable intrusion detection system based on deep learning. *International Journal of Information Security* **2021**, *20*. https://doi.org/10.1007/s10207-020-00508-5.
23. Odeh, A.; Abu Taleb, A. Ensemble-Based Deep Learning Models for Enhancing IoT Intrusion Detection. *Applied Sciences* **2023**, *13*, 11985. Available online: https://doi.org/10.3390/app132111985.
24. Qin, J.; Wang, J.; Lei, T.; Sun, G.; Yue, J.; Wang, W.; Chen, J.; Qian, G. Deep learning-based software and hardware framework for a noncontact inspection platform for aggregate grading. *Measurement* **2023**, *211*, 112634. Available online: https://doi.org/10.1016/j.measurement.2023.112634.
25. Talukder, M.A.; Islam, M.; Uddin, M.A.; Hasan, F.; Sharmin, S.; Alyami, S.; Moni, M.A. Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. *Journal of Big Data* **2024**, *11*, In Press. Available online: https://doi.org/10.1186/s40537-024-00886-w.
26. Saba, T.; Rehman, A.; Sadad, T.; Kolivand, H.; Bahaj, S. Anomaly-based intrusion detection system for IoT networks through deep learning model. *Computers & Electrical Engineering* **2022**, *99*, 107810. Available online: https://doi.org/10.1016/j.compeleceng.2022.107810.
27. Shone, N.; Tran Nguyen, N.; Vu Dinh, P.; Shi, Q. A Deep Learning Approach to Network Intrusion Detection. *IEEE Transactions on Emerging Topics in Computational Intelligence* **2018**, *2*, 41–50. https://doi.org/10.1109/TETCI.2017.2772792.
28. Mbow, M.; Koide, H.; Sakurai, K. An Intrusion Detection System for Imbalanced Dataset Based on Deep Learning. *IEEE Computer Society Annual Symposium on Candar (CANDAR)* **2021**, 38–47. https://doi.org/10.1109/CANDAR53791.2021.00013.
29. Rajadurai, H.; Gandhi, U. A stacked ensemble learning model for intrusion detection in wireless network. *Neural Computing and Applications* **2022**, *34*. https://doi.org/10.1007/s00521-020-04986-5.
30. Thirimanne, S.; Jayawardana, L.; Liyanaarachchi, P.; Yasakethu, L. Comparative Algorithm Analysis for Machine Learning Based Intrusion Detection System. *IEEE International Conference on Information and Automation for Sustainability (ICIAfS)* **2021**, 191–196. https://doi.org/10.1109/ICIAfS52090.2021.9605814.
31. Alotaibi, Y.; Ilyas, M. Ensemble-Learning Framework for Intrusion Detection to Enhance Internet of Things' Devices Security. *Sensors* **2023**, *23*, 5568. Available online: https://doi.org/10.3390/s23125568.
32. Mhawi, D.N.; Hashem, S.; Aldallal, A. Advanced Feature-Selection-Based Hybrid Ensemble Learning Algorithms for Network Intrusion Detection Systems. *Symmetry* **2022**, *14*, 1461. Available online: https://doi.org/10.3390/sym14071461.
33. Almomani, A.; Akour, I.; Manasrah, A.; Almomani, O.; Alauthman, M.; Abdullah, E.; Shwait, A.; Al, R. Ensemble-Based Approach for Efficient Intrusion Detection in Network Traffic. *Intelligent Automation and Soft Computing* **2023**, *37*, 2499–2517. Available online: https://doi.org/10.32604/iasc.2023.039687.
34. Urmi, W.F.; Uddin, M.N.; Uddin, M.A.; Talukder, M.A.; Hasan, M.R.; Paul, S.; Chanda, M.; Ayoade, J.; Khraisat, A.; Hossen, R.; Imran, F. A stacked ensemble approach to detect cyber attacks based on feature selection techniques. *International Journal of Cognitive Computing in Engineering* **2024**, *5*, 316–331. Available online: https://doi.org/10.1016/j.ijcce.2024.07.005.
35. Kumari, T.A.; Mishra, S. Tachyon: Enhancing stacked models using Bayesian optimization for intrusion detection using different sampling approaches. *Egyptian Informatics Journal* **2024**, *27*, 100520. Available online: https://doi.org/10.1016/j.eij.2024.100520.
36. Li, Y.; Xu, Y.; Liu, Z.; Hou, H.; Zheng, Y.; Xin, Y.; Zhao, Y.; Cui, L. Robust Detection for Network Intrusion of Industrial IoT Based on Multi-CNN Fusion. *Measurement* **2019**, *154*, 107450. https://doi.org/10.1016/j.measurement.2019.107450.
37. Mambwe Sydney, K. A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework. *Computer Communications* **2022**, *199*. https://doi.org/10.1016/j.comcom.2022.12.010.
38. Ros, S.; Eang, C.; Tam, P.; Kim, S. ML/SDN-Based MEC Resource Management for QoS Assurances. In *Proceedings of the [Conference Name]* **2023**, 591–597. ISBN 978-981-99-1251-3. https://doi.org/10.1007/978-981-99-1252-0_79.
39. Peng, H.; Shen, X. Multi-Agent Reinforcement Learning Based Resource Management in MEC- and UAV-Assisted Vehicular Networks. *IEEE Journal on Selected Areas in Communications* **2020**, *PP*, 1–1. https://doi.org/10.1109/JSAC.2020.3036962.
40. Latah, M.; Toker, L. Minimizing false positive rate for DoS attack detection: A hybrid SDN-based approach. *ICT Express* **2019**, *6*. https://doi.org/10.1016/j.icte.2019.11.002.
41. Santos, R.; Silva, D.; Santo, W.; Ribeiro, A.; Ordonez, E. Machine learning algorithms to detect DDoS attacks in SDN. *Concurrency and Computation: Practice and Experience* **2019**, *32*, e5402. https://doi.org/10.1002/cpe.5402.

42. Amouri, A.; Alaparthy, V.; Morgera, S. A Machine Learning Based Intrusion Detection System for Mobile Internet of Things. *Sensors* **2020**, *20*. https://doi.org/10.3390/s20020461.
43. Khan MSI, Rahman A, Debnath T, et al. Accurate brain tumor detection using deep convolutional neural network. *Computational and Structural Biotechnology Journal* 2022; 20: 4733–4745.
44. Song, H.; Woo, J.; Kim, H. K. In-vehicle network intrusion detection using deep convolutional neural network. *Vehicular Communications* **2019**, *21*, 100198. doi:10.1016/j.vehcom.2019.100198.
45. Malhotra, H.; Sharma, P. Intrusion Detection using Machine Learning and Feature Selection. *Int. J. Comput. Netw. Inf. Secur.* **2019**, *11*, 43–52. doi:10.5815/ijcnis.2019.04.06.
46. Anichur Rahman, Md Saikat Islam Khan, Antonio Montieri, Md Jahidul Islam, Md Razaul Karim, Mahedi Hasan, Dipanjali Kundu, Mostofa Kamal Nasir, and Antonio Pescapè. *BlockSD-5GNet: Enhancing security of 5G network through blockchain-SDN with ML-based bandwidth prediction*. Transactions on Emerging Telecommunications Technologies, 35(4):e4965, 2024, Wiley Online Library.
47. Liu, H.; Lang, B. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. *Applied Sciences* **2019**, *9*, 4396. doi:10.3390/app9204396.
48. Md Anwar Hussen Wadud, Anichur Rahman, Sadia Sazzad, Dipanjali Kundu, Muaz Rahman, Airin Afroj Aishi, Sm Nuruzzaman Nobel, TM Amir Ul Haque Bhuiyan, and Zobayer Ahmed, “Garduino: Sustainable Indoor Gardening Developed with Mobile Interface,” *Statistics, Optimization & Information Computing*, 2024.
49. Pinto, A.; Herrera, L.-C.; Donoso, Y.; Gutierrez, J. Survey on Intrusion Detection Systems Based on Machine Learning Techniques for the Protection of Critical Infrastructure. *Sensors* **2023**, *23*, 2415. doi:10.3390/s23052415.
50. Lin, W.; Wu, Z.; Lin, L.; Wen, A.; Li, J. An Ensemble Random Forest Algorithm for Insurance Big Data Analysis. *IEEE Access* **2017**, *PP*, 1-1. doi:10.1109/ACCESS.2017.2738069.
51. Abubakar, A.; Pranggono, B. Machine learning based intrusion detection system for software defined networks. *IEEE Int. Conf. Emerg. Secur. Technol. (EST)* **2017**, 138–143. doi:10.1109/EST.2017.8090413.
52. Meti, N.; Narayan, D. G.; Baligar, V. Detection of distributed denial of service attacks using machine learning algorithms in software defined networks. *IEEE Int. Conf. Adv. Comput. Commun. Informat. (ICACCI)* **2017**, 1366–1371. doi:10.1109/ICACCI.2017.8126031.
53. Tang, T.; Mhamdi, L.; McLernon, D.; Zaidi, S. A. R.; Ghogho, M. Deep learning approach for network intrusion detection in software defined networking. *Proc. IEEE/ACS 13th Int. Conf. Comput. Syst. Appl. (AICCSA)* **2016**. doi:10.1109/WINCOM.2016.7777224.
54. Sharma, S.; Gigras, Y.; Chhikara, R.; Dhull, A. Analysis of NSL KDD dataset using classification algorithms for intrusion detection system. *Recent Patents Eng.* **2018**, *12*. https://doi.org/10.2174/1872212112666180402122150doi: 10.2174/1872212112666180402122150.
55. Natras, R.; Soja, B.; Schmidt, M. Ensemble machine learning of random forest, AdaBoost and XGBoost for vertical total electron content forecasting. *Remote Sensing* **2022**, *14*, 3547. https://www.mdpi.com/2072-4292/14/15/3547doi:10.3390/rs14153547.
56. Cynthia Anthony.; Walid Elgenaidi.; Muzaffar Rao. Intrusion Detection System for Autonomous Vehicles Using Non-Tree Based Machine Learning Algorithms. *electronics* **2024**, *5*, 809. https://www.mdpi.com/2079-9292/13/5/809doi.org/10.3390/electronics13050809.
57. Thockchom, Ngamba and Singh, Moirangthem Marjit and Nandi, Utpal A novel ensemble learning-based model for network intrusion detection. *Springer* **2023**, *9*, 5. https://link.springer.com/article/10.1007/s40747-023-01013-7 doi.org/10.1007/s40747-023-01013-7.