International Journal of Innovative Research and Scientific Studies, 8(1) 2025, pages: 1964-1978

IJIRSS
International Journal of Innovative Research & Scientific Studies

ISSN: 2617-6548

URL: www.ijirss.com

Machine learning - powered intrusion detection system for agriculture 4.0: Securing the next generation of farming

Udaya Kumar Addanki[1*], Ravi Babu Devareddi[2], Kishore Kumar Kamarajugadda[3], Movva Pavani[4], Pradeepini Gera[5]

[1]*Department of Artificial Intelligence, School of Engineering, Anurag University, Hyderabad, India.*
[2]*Department of CSE, GITAM School of Technology, GITAM Deemed to be University, Hyderabad, India.*
[3]*Department of ECE, The ICFAI University, Raipur, Chattisgarh, India.*
[4]*Department of ECE, Nalla Malla Reddy Engineering College, Divyanagar, Hyderabad, India.*
[5]*Department of CSE, School of Computing, Koneru Lakshmaih Education Foundation, Deemed to be University, Vijayawada, AP, India.*

Corresponding author: Udaya Kumar Addanki (*Email: udayaka.18@gmail.com*)

Abstract

Agriculture is a fundamental component of India's economy, involving more than half of the workforce and significantly contributing to revenue generation via exports. Integrating IoT, drones, and artificial intelligence within Agriculture 4.0 transforms conventional farming methods, enhancing productivity and sustainability. However, the integration of IoT devices in open fields poses considerable cybersecurity issues, including susceptibility to Distributed Denial of Service (DDoS) attacks and the potential for data manipulation. These assaults undermine the integrity, reliability, and safety of agricultural systems, underscoring the pressing need for stringent security measures. Recent research underscores the efficacy of intrusion detection systems (IDS) in detecting and mitigating these attacks. This study tackles existing gaps by introducing a meta-model that integrates XGBoost (XG), Random Forest (RF), Decision Trees (DT), and a meta-learner utilizing XGBoost. The model demonstrated impressive results, achieving an accuracy of 96.37%, precision of 96.39%, recall of 96.37%, and an F1 score of 96.25%. These findings illustrate the model's effectiveness in ensuring Agriculture 4.0, which contributes to resilient and sustainable innovative farming ecosystems.

Keywords: Agriculture 4.0, Cybersecurity, DDoS, Intrusion detection systems, IoT, Machine learning, Meta-model, Smart farming, sustainable agriculture, XGBoost.

DOI: 10.53894/ijirss.v8i1.4840
Funding: This study received no specific financial support.
History: Received: 10 January 2025/**Revised:** 12 February 2025/**Accepted:** 14 February 2025/**Published:** 21 February 2025

Competing Interests: The authors declare that they have no competing interests.
Authors' Contributions: All authors contributed equally to the conception and design of the study. All authors have read and agreed to the published version of the manuscript.
Transparency: The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.
Publisher: Innovative Research Publishing

1. Introduction

Agriculture is very important for India's economy, providing jobs for many people and generating significant revenue through exports. Agriculture 4.0 refers to the use of technology such as IoT, drones, and AI to enhance farming practices. However, there's a significant problem. With all these advanced gadgets being used in the fields, it's like leaving the door open for cyber attackers to misuse them. They could potentially implement actions such as shutting down essential services or introducing fake data. This could disrupt food safety and the efficiency of food distribution from farms to our plates. In terms of evolution in agriculture, Agriculture 1.0 is the earliest form of agriculture, where everything was done manually. Farmers relied on traditional methods such as hand tools, animal power, and natural fertilizers. It was a labor-intensive process with low productivity and limited scope for large-scale farming, as shown in Figure 1.

Farming significantly improved in Agriculture 2.0, and farmers began using large machines like tractors, plows, and harvesters. These machines made it easier to plant seeds, till the soil, and harvest crops. Additionally, they started using chemicals to help their crops grow better and to keep harmful insects and weeds away. These chemicals, called fertilizers and pesticides, enabled farmers to grow more food on the same amount of land. Essentially, Agriculture 2.0 was all about using machines and chemicals to make farming faster, easier, and more productive.

Agriculture 3.0 saw the introduction of technology-driven farming practices. It involved using early forms of automation, such as irrigation systems and crop monitoring devices. As technology improved, farming became even more high-tech. Farmers began using devices like sprinklers to water their crops automatically and sensors to monitor soil and plants [1].

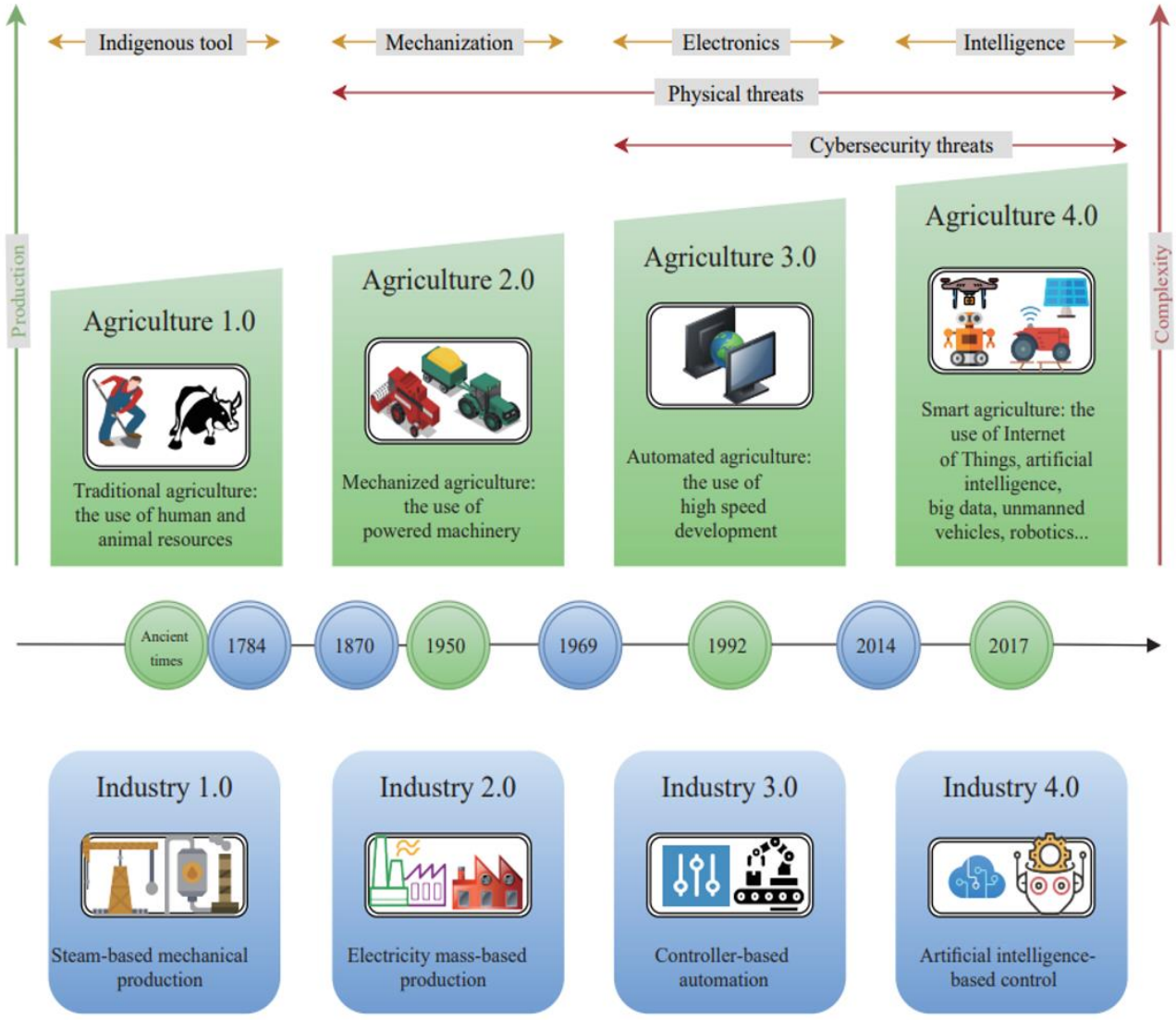

Figure 1.
Illustrate the evolution of agricultural revolutions alongside industrial revolutions and the associated cybersecurity threats.
Source: Ferrag, et al. [2]

Agriculture 4.0, as described by Aceto, et al. [3]; Industry 4.0 and cybersecurity [4] and Ferrag, et al. [5], represent the current generation, incorporating modern technologies into all aspects of farming. It comprises IoT sensors for real-time monitoring of soil moisture, soil temperature, and crop health. Drones are used for aerial photography and precision agriculture, allowing for more targeted resource allocation [6]. AI and machine learning systems analyze enormous volumes of data to simplify farming procedures, estimate crop harvests, and improve resource management. Blockchain technology

is also being considered for supply chain traceability and transparency [7]. Agriculture 4.0 seeks to increase production while reducing resource consumption and environmental effects, ushering in a new era of smart and sustainable farming.

This study describes the advancement of an effective IDS to address cybersecurity issues in Agriculture 4.0. The IDS uses machine-learning techniques to safeguard agricultural systems and discover and mitigate network vulnerabilities. The system identifies and prevents unauthorized access attempts by monitoring network data for signs of malicious activity [8]. Classifiers, or prediction models, are used to distinguish between harmful and lawful associations [9]. This strategy enhances Agriculture 4.0's resilience by maintaining essential agricultural infrastructure and ensuring that food production and distribution networks continue to operate without interruption [10].

2. Literature Survey

2.1. Survey on Intrusion Detection System

The advancement of harmful malware Disha and Waheed [11] presents a substantial obstacle to intrusion detection systems (IDS). High-profile cybercrime incidents highlight the global spread of cyber threats, emphasizing the need for strict computer security measures.

Khraisat, et al. [12] examined signature-based and anomaly-based techniques to determine their ease of use, efficacy, and impact on intrusion detection systems. Unlike earlier surveys, this one covers IDS datasets, strategies, assessment methodologies, and varied attack types. This study organizes IDSs by signature detection (identifying known threat patterns) and anomaly detection (detecting anomalous activity) rather than by detection techniques or attack types. The research also underlines the importance of datasets used to train and evaluate IDSs on system performance. It emphasizes machine learning and deep learning to build IDSs that can identify and mitigate contemporary attacks. The authors warn that these methods may produce false positives (identifying harmless behavior as a threat) or false negatives (missing real dangers). The report also highlights the need for updated assessment methodologies since many current tests use outdated datasets that do not reflect modern attackers' techniques.

Dutt, et al. [13] proposed a model for IDS that adopts a biologically inspired approach, mirroring the functionality of the human immune system with its two-layered architecture. This work introduces a new IDS inspired by the human immune system. It consists of two components, similar to the human immune system: statistical modeling anomaly detection (SMAD), which detects initial flaws in network traffic, and adaptive immune-based anomaly detection (AIAD), which looks for unusual packets. They were tested using real-time and standard datasets, showing high accuracy rates, up to 96.04% with real-time traffic and nearly 99% with standard datasets. The first layer (SMAD) captures suspicious network traffic, achieving high true positive rates. The second layer (AIAD) analyzes these packets, achieving high accuracy rates. Tests on real-time and standard datasets showed effective intrusion detection capabilities.

Rose, et al. [14] proposed a dataset and model for detecting possible threats in IoT devices by combining network characterization and machine learning to mitigate cyberattacks. Their approach employs anomaly-based intrusion detection system (IDS) profiles that aggressively monitor all devices in real-time for tampering attempts. Furthermore, the technology identifies fraudulent network transactions, ensuring comprehensive security for IoT devices.

Thakkar and Lohiya [15] looked at datasets that are used for machine learning (ML) and data mining-based intrusion detection systems (IDS). They emphasized the importance of updating these datasets to accurately detect recent attacks, as attackers constantly use new methods and technologies. The study stressed the need for datasets that reflect real-world network scenarios, prompting the introduction of the CICD-2017 and CICIDS-2018 datasets. These new datasets provide realistic network traffic and updated attack patterns for more effective intrusion detection.

These papers collectively emphasize the need for advanced IDS to detect and overcome modern cyber threats, including malware and attacks on IoT devices. The research highlights the use of innovative techniques like ML and anomaly detection to enhance the effectiveness of IDS. Additionally, updating datasets and testing methodologies ensures IDSs can effectively adapt to evolving cyber threats.

2.2. Survey on IDS using ML

Amar and Bouabid [16] developed a hybrid IDS employing ML techniques, including KNN, NB, LR, and SVM trained on the NSL-KDD dataset. Results showed that KNN exhibits superior performance compared to NB, RF, and SVM for binary and multiclass classification tasks, achieving a precision factor of 93.28% and an accuracy of 96.69%. This emphasizes KNN's effectiveness in detecting intrusions within the IDS framework.

Bertoli, et al. [17] presented the AB-TRAP architecture, which enables the full implementation of network traffic and operational factors to improve security. This technique includes creating datasets for both harmful and lawful traffic, training machine learning models, deploying them on target systems, and evaluating the security module's efficacy. The Decision Tree (DT) generated the best outcomes of the models tested.

According to Magán-Carrión, et al. [18] four machine-learning models can be employed to detect system threats. These models are the random forest, the support vector classifier with a linear kernel (SVC-L), and the support vector classifier with a radial basis function kernel (SVC-RBF). Their research focused on four different sorts of attacks: scanning, spamming, denial of service (DoS), and botnets. The outcomes of this study are crucial for developing more powerful network intrusion detection systems.

Urda, et al. [19] proposed IDS architecture to address new challenges in managing large network traffic in real-time settings. Its technique employs the optimal allocation methodology, which chooses representative samples from a given dataset in the first instance. These are then used in the SVM model to solve classification problems. This method was tested on anomaly detection using the KDDCup'99 dataset. A feature selection method that combines human exploratory feature

analysis with two feature selection methods, forward selection ranking and backward selection ranking, was also suggested. This method produced the best results for modeling and detection time.

Prasad, et al. [20] introduced an efficient feature selection approach that analyzes and ranks features using anticipated probabilities. This method can produce predictable probabilities, with greater probability suggesting more significant features and zeros indicating ineffectual traits, filling the dataset with unnecessary information. The proposed hybrid system improves detection capabilities while decreasing false alarm rates by merging two well-known machine learning techniques: RST and Bayes' theorem [20]. This method increases learning efficiency by designing a function based on rough-set approximations that use Bayes' theorem to deal with uncertainty. This allows the system to detect regular and unexpected packets, as well as determine what weird behavior occurs.

Megantara and Ahmad [21] created a hybrid intrusion detection model that combines supervised feature optimization with unsupervised data reduction approaches. The technique uses an attribute importance decision tree (DT) mechanism combined with recursive feature removal to identify relevant attributes. The Local Outlier Factor (LOF) technique identifies anomalous or outlier data. The NSL-KDD and UNSW-NB15 datasets were utilized to evaluate the proposed model, and it outperformed earlier models.

Wang, et al. [22] simplified the interpretation of IDS by employing Shapley additive explanations (SHAPs), which combine local and global explanations. While the testing framework makes use of the NSL-KDD dataset, the local explanation explains why the model made certain conclusions for specific inputs.

Saranya, et al. [23] address the growing security challenges of increased Internet attacks amid rapid technological advancements. It focuses on strengthening information security by integrating machine learning (ML) algorithms into IDS. The study conducts a comparative analysis of ML algorithms like LDA, CART, and RF for intrusion classification in applications like fog computing, IoT, big data, smart cities, and 5G networks. Notably, the DT model demonstrates superior performance.

Soumya and Kapil [24] highlight IoT devices' exponential growth and the subsequent increase in their vulnerability to cyber-attacks. Hackers exploit these vulnerabilities to launch damaging attacks to drain resources from IoT networks. To protect against these kinds of threats, the method uses machine learning with carefully chosen datasets to find intrusions in IoT networks. The process begins with presenting various intrusion detection datasets, then utilizing the IoTID20 dataset and identifying classification features, and then extracting significant characteristics from the dataset and applying ML methods such as linear algorithms, random forests, and gradient boost algorithms to identify anomalies within the IoT network accurately.

Nabi and Zhou [25] research focuses on making automated intrusion detection more accurate and efficient by creating a classifier that can detect intrusions with minimal false alarms. They are motivated to address the challenge of dealing with intrusion detection data and improve how well classifiers can spot intrusions. They used the NSL-KDD dataset, which has a bunch of abnormal and regular network traffic samples for training and testing. At first, they used all the features in the dataset to train the classifiers, and the results were good. The J48 tree classifier did the best, with an accuracy of 79.1 percent. Then, they tried two techniques, random projection, and PCA, to improve the classifiers. Random Projection was effective, especially with the PART algorithm, which achieved an accuracy of 82.0%. It also saved time compared to PCA. Research shows that random projection can make intrusion detection more accurate and faster, a big deal for cybersecurity.

Azizan, et al. [26] compare three ML algorithms, DJ, RF, and SVM, for their efficacy in identifying abnormalities in network traffic, which is critical for IDS. The KDD approach and the CICIDS 2017 dataset are used to develop and evaluate an ML-based NIDS. The SVM method is the most accurate, with an average accuracy of 98.18% and a maximum precision of 98.74%. However, RF outperforms SVM in the average recall, at 97.62%. Overall, people choose SVM as the best intrusion detection algorithm due to its high accuracy and precision, which offer valuable insights for enhancing system security.

Researchers investigated a variety of machine-learning techniques for improving intrusion detection systems (IDS) for network security utilizing datasets such as NSL-KDD and CICIDS 2017. Amar et al. discovered that K-Nearest Neighbours (KNN) was effective on NSL-KDD, but Gustavo et al. proposed AB-TRAP with a decision tree, which showed potential. Magán-Carrión, et al. [18] evaluated models at UGR'16, Urda, et al. [19] proposed LS-SVM at KDDCup'99, and Prasad, et al. [20] used RST and Bayes' theorem. Megantara and Ahmad [21] combined supervised and unsupervised approaches to improve accuracy, while Wang, et al. [22] employed SHAPs for interpretation. Saranya, et al. [23] discovered that decision trees work well for diverse networks, while Soumya and Kapil [24] examined IoT security. In the NSL-KDD focus, J48 achieved 79.1% accuracy, which was increased to 82.0% using random projection. Finally, SVM demonstrated good accuracy (98.18%) and precision (98.74%) on CICIDS2017, whilst RF demonstrated high recall (97.62%). These studies strive to improve intrusion detection systems to enhance network security.

2.3. Survey on Papers using DL for IDS

Aldhyani and Alkahtani [27] proposed a DL-based intrusion detection system (IDS) to safeguard IoT networks, particularly in Agriculture 4.0. Utilizing LSTM and CNN-LSTM architectures, the system effectively detects DDoS attacks with precision. Leveraging the CIC-DDoS2019 dataset, the proposed model achieves 100% precision across all evaluation metrics.

Ho, et al. [28] proposed that CNN could distinguish cyberattacks in IDS. Evaluating the CICIDS 2017 dataset with CNN yielded an accuracy rate of 94.96%. Furthermore, it identified new DoS cases not encountered during training.

Sun, et al. [29] devised the CNN-LSTM technique, which combines convolutional neural networks (CNN) and long short-term memory (LSTM) networks to enhance intrusion detection efficiency. The improved model was tested using the

CICIDS dataset and achieved 98.7% accuracy. However, the model's detection accuracy is limited for some attack types, such as Heartbleed and SSH-Patator.

Di Mauro, et al. [30] compared neural-based approaches, focusing on artificial neural networks (ANN). They used the CICIDS 2017 and KDD99 datasets to create ANN-based models. While ANN approaches performed well overall, the backpropagation strategy resulted in poor processing speeds. It was hypothesized that omitting the feature optimization phase would reduce the time complexity of these classifiers.

Hidayat, et al. [31] employed machine learning techniques and a unique hybrid feature selection method that combines the Pearson correlation coefficient and Random Forest models to improve intrusion detection capabilities. They tested models like LSTM, KNN, decision trees (DT), and AdaBoost on the TON_IoT dataset. Their research emphasized the merits and limitations of these approaches for detecting novel attack types.

Kasongo [32] designed an IDS using Recurrent Neural Networks (RNN), specifically LSTM, Gated Recurrent Units (GRU), and basic RNN architectures. The NSL-KDD and UNSW-NB15 datasets were chosen for analysis due to their significance in intrusion detection research. The study also examined decision trees (DT), random forest (RF), and XGBoost (XGB) models for implementing IDS on software-defined networks (SDN). XGBoost outperformed the other models, achieving the highest F1 score, accuracy, and recall [33].

Researchers have been developing various systems to detect unauthorized access attempts to computer networks using machine learning. For instance, Aldhyani and Alkahtani developed a system that employs LSTM, a type of deep learning, and CNN-LSTM, achieving 100% accuracy in detecting attacks on the CIC-DDoS2019 dataset. Samson and their team used a method called CNN to identify attacks, achieving 94.96% accuracy on the CICIDS 2017 dataset, including new ones they had not encountered before. Sun and their group enhanced CNN by incorporating LSTM to improve attack detection. They achieved 98.7% accuracy on the CICIDS 2017 dataset, although some attacks were more challenging to detect. Mario and their team compared different methods for detecting attacks and favored ANN, despite its slower performance. Additionally, a study employed a combination of machine learning and feature extraction to enhance the accuracy of IDS in detecting attacks, finding that decision trees and MLPs were the most accurate against new types of attacks. Sydney Mambwe Kasongo developed a system using various types of RNNs, such as LSTM and Simple RNN, which performed well in detecting attacks with the help of a method called XGBoost on the UNSW-NB15 and NSL-KDD datasets. Their XGBoost-LSTM system achieved 88.13% accuracy in identifying attacks in NSL-KDD, demonstrating its effectiveness in detecting attacks in diverse scenarios.

2.4. Survey on Papers using ML and DL for IDS

Alzahrani and Alenazi [34] showed that deep learning algorithms with LSTM networks and GRU, such as GANs and autoencoders, can effectively identify changes in networks. These approaches are adept at capturing complex patterns within data, thus enabling IDSs to adapt to various types of threats in SDN environments that evolve through network traffic.

El-Sayed, et al. [35] evaluated seven supervised learning approaches of varying complexity. They classified these algorithms into two categories: CNN-based classifiers (such as two-layer CNN, four-layer CNN, and VGG16) and classical classifiers (such as Logistic Regression (LR), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN)). Their findings indicated that the SVM technique, which incorporates features collected with MobileNetv2, outperformed the other models, achieving a 94% accuracy rate. This higher performance is attributed to SVM's efficient and consistent training technique, which requires less computational power.

Ferrag, et al. [5] proposed a lightweight random neural network architecture to detect cyber vulnerabilities in the Industrial Internet of Things (IIoT). They tested the proposed method on the DS2OS open-source dataset and discovered that it outperformed well-known machine learning approaches, including support vector machines (SVM), decision trees (DT), and artificial neural networks. The DS2OS dataset includes seven distinct types of cyberattacks: denial of service (DoS), scanning, data type probing, malicious control, incorrect configuration, espionage, and malicious actions. This study highlights the effectiveness of random neural networks in detecting cyber threats in IIoT environments.

Disha and Waheed [11] studied various learning algorithms for intrusion detection, including Decision Tree (DT), Gradient Boosted Tree (GBT), AdaBoost, Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU). The models were tested on the UNSW-NB15 and Network TON datasets, with improvements made using the GIWRF feature selection method. The decision tree classifier achieved the highest accuracy among the models tested. However, the study was limited to binary classification and did not consider multiclass classification settings, indicating a need for further research.

Priyanka and Gireesh Kumar [36] proposed various approaches. The RFs, NBs, and CNNs methods were divided into two categories: two-class and multiclass. It was observed that NB had low accuracy rates in both classification problems and poor performance during multiclass categorization tasks.

Thapa, et al. [37] emphasized the importance of robust anomaly-based network detection systems in enhancing cybersecurity by responding to emerging and evolving threats with better flexibility and agility. Traditional intrusion detection systems (IDS) often struggle to maintain high detection accuracy while minimizing false alarms. The authors developed an IDS framework that combines ML and DL models to address these issues. They then tested it against the CIDDS dataset. The paper evaluates the performance of individual ML and DL models and proposes using an ensemble approach to improve detection metrics by integrating the best-performing models. Benchmarking findings against the CIC-IDS2017 dataset illustrate CIDDS's efficacy, as it combines current attack patterns and typical usage scenarios within a simulated office setting. The study emphasizes the need to balance accuracy and interpretability when developing AI-driven IDS solutions.

Researchers have been enhancing IDS to protect networks from cyber threats. Abdul Salam et al. discovered that XGBoost was most effective for detecting intrusions in software-defined networks using the NSL-KDD dataset. They also recommended deep learning methods like autoencoders and recurrent neural networks (RNNs) to better detect network changes. El-Sayed et al. compared different algorithms and found the Support Vector Machine (SVM) to be the most accurate, achieving 94% accuracy on MobileNetv2 features. Latif et al. used the DS2OS dataset to create a lightweight random neural network that outperforms standard approaches for identifying cyber risks in industrial IoT. Raisa et al. identified the decision tree as the top performer among learning algorithms for intrusion detection. Priyanka et al. developed IDS approaches utilizing the CICIDS 2017 dataset, and Random Forest outperformed the Convolutional Neural Network (CNN). Another study employed machine learning and deep learning models to create a reliable anomaly-based detection system with high accuracy and minimal false alarms using Coburg intrusion detection datasets (CIDDSs). Their ensemble model combines the most effective ML and DL models to enhance performance. These efforts aim to improve network security by effectively detecting and preventing attacks.

3. Methodology

Designing an efficient Intrusion Detection System (IDS) entails many essential phases, each critical to guaranteeing the system's efficacy and precision in detecting and responding to cybersecurity threats. These steps include:

- Data Collection: The initial stage in the IDS process is to obtain network traffic data for analysis. This data may contain a variety of characteristics, such as the service type, flag, source bytes, protocol type, and destination bytes, among others. The quality and relevance of the data primarily determine the effectiveness of the IDS.
- Preprocessing: Data quality and relevancy substantially influence an IDS's efficacy. Preprocessing prepares data for model training through cleaning, normalizing, encoding, and selecting features. These processes ensure that the data is accurate and appropriate, enhancing the IDS's performance.
- Feature Selection: aims to discover the most significant characteristics contributing to intrusion detection, reducing data dimensionality and decreasing overfitting. By selecting the appropriate attributes, the IDS may become more efficient and precise in detecting potential threats.
- Model Training: Machine learning models are trained using preprocessed and feature-selected data during this step. The training models utilized in this study include XGBoost and Random Forest [38]. Each model learns to distinguish between regular network traffic and potentially malicious traffic using the data it receives for training.
- Intrusion detection: After training, models can monitor network traffic. The models analyze the incoming network traffic and predict whether it is a normal or a potential threat. This is the core function of the IDS.
- Alerts: If a potential threat is detected, the IDS doesn't just stop at detection; it also alerts the network administrator and logs the event for further investigation. This allows for quick action to be taken to mitigate the threat.

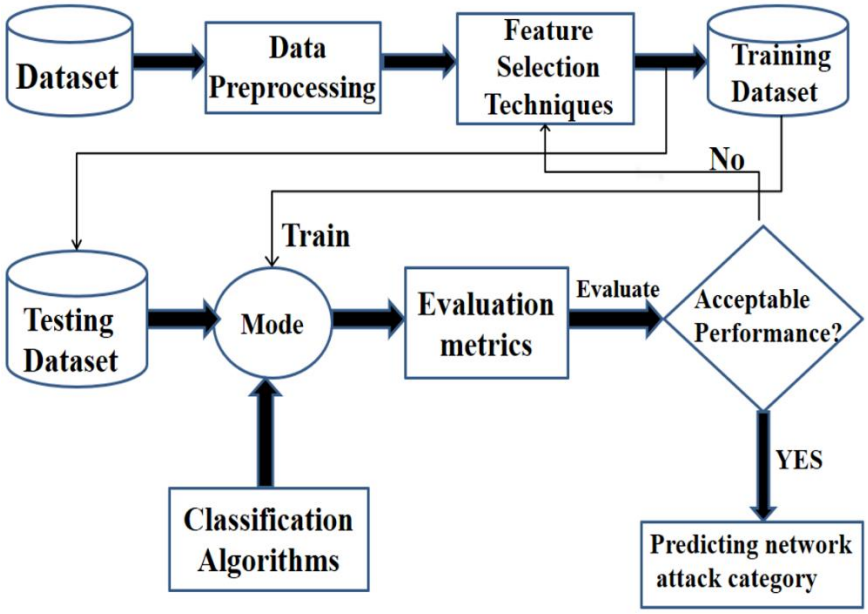

Figure 2.
Content Diagram.

Figure 2 illustrates the preprocessing and feature selection processes applied to the dataset to generate a training dataset. A classification algorithm undergoes training and evaluation through performance metrics, with retraining implemented if performance is deemed unsatisfactory. Upon achieving acceptable performance, the model accurately predicts the category of network attacks.

3.1. Data Collection

Datasets from Kaggle provide information across various industries. Several datasets, such as the KDD99-Cup dataset, UNSW-NB15, NSL-KDD, NGIDS-DS, etc., are used to work on intrusion detection systems. The main drawback is that because these databases are outdated, they cannot assist in identifying every kind of newly emerging threat. Additionally, data is taken from Reddit.

Figure 3 illustrates that the Normal category has the greatest frequency, followed by Generic, Exploits, and Fuzzers as the predominant attack types. Denial of Service, Reconnaissance, and Analysis assaults occur with moderate frequency, while Backdoor, Shellcode, and Worms are the least prevalent. This distribution underscores the prevalence of Generic and Exploit assaults, emphasizing the need for robust cybersecurity solutions.

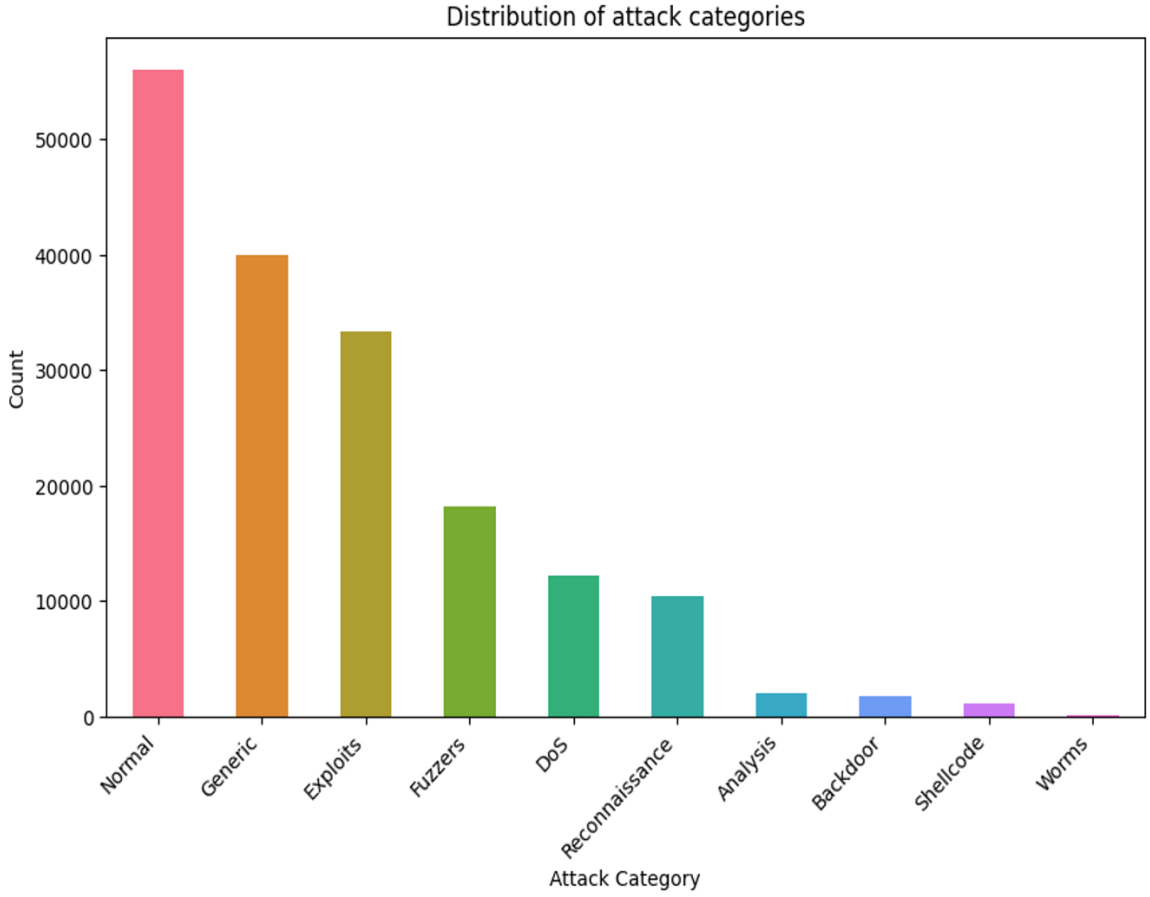

Figure 3.
Distribution of attacks.

3.2. Data Preprocessing

Data preprocessing is the systematic cleansing and preparation of datasets to eliminate extraneous or noisy information, ensuring correctness and dependability in future studies. Effective management is essential since oversight may result in erroneous findings. This work used the UNSW-NB15 dataset, which has 175,341 occurrences and 45 characteristics, requiring comprehensive preprocessing to make it suitable for analysis.

Key preprocessing stages included the elimination of punctuation, null values, and noise, which were suited to the project's unique requirements. The dataset was refined using ordinal encoding and methods for removing null values.

Ordinal encoding is one of the techniques used in data preprocessing to convert categorical variables into numerical values. Ordinal encoding has two major advantages:

1. Algorithm compatibility involves transforming categorical data into a numerical representation suitable for machine learning algorithms, thereby facilitating more efficient mathematical computations. One-hot, label, and binary encoding are prominent techniques for this modification, allowing models to learn from the input [39] successfully.
2. Preservation of ordinal relationships: Maintains the intrinsic order or hierarchy between categories, which can sometimes be lost when approaches such as one-hot encoding are employed.

Ordinal encoding assigns a unique numerical value to each distinct category, typically starting from zero and incrementing by one for each subsequent category. Consider the categorical variable "size" with values ["small," "medium," "large"]. These categories would be translated into numeric values by Liu, et al. [1] and Ferrag, et al. [2], while retaining their natural order using ordinal encoding. This method is advantageous when the ordinal connection between categories is relevant to the study or model.

3.3. Feature Selection

It improves model performance by identifying the most important features, reducing dimensionality, and removing unnecessary or duplicate information. This technique enhances model accuracy, decreases overfitting, and boosts computational efficiency. Additionally, it improves prediction accuracy, eliminates overfitting, and accelerates training. There are several methods for feature selection, including:

- Correlation-based FS: feature selection involves selecting features that have the highest correlation with the target variable.
- Spearman correlation is a statistical strategy for determining the degree and direction of a link between two variables based on their ranking order. Unlike Pearson's correlation, which analyzes linear connections, Spearman's correlation assesses monotonicity, which occurs when one variable consistently increases or decreases in tandem with another. This approach is especially effective for ordinal data or when the normality and linearity conditions of Pearson's correlation are not satisfied.

3.4. Model Training

The three main algorithms used in building Intrusion Detection Systems (IDS) are XGBoost, CatBoost, and Random Forest.

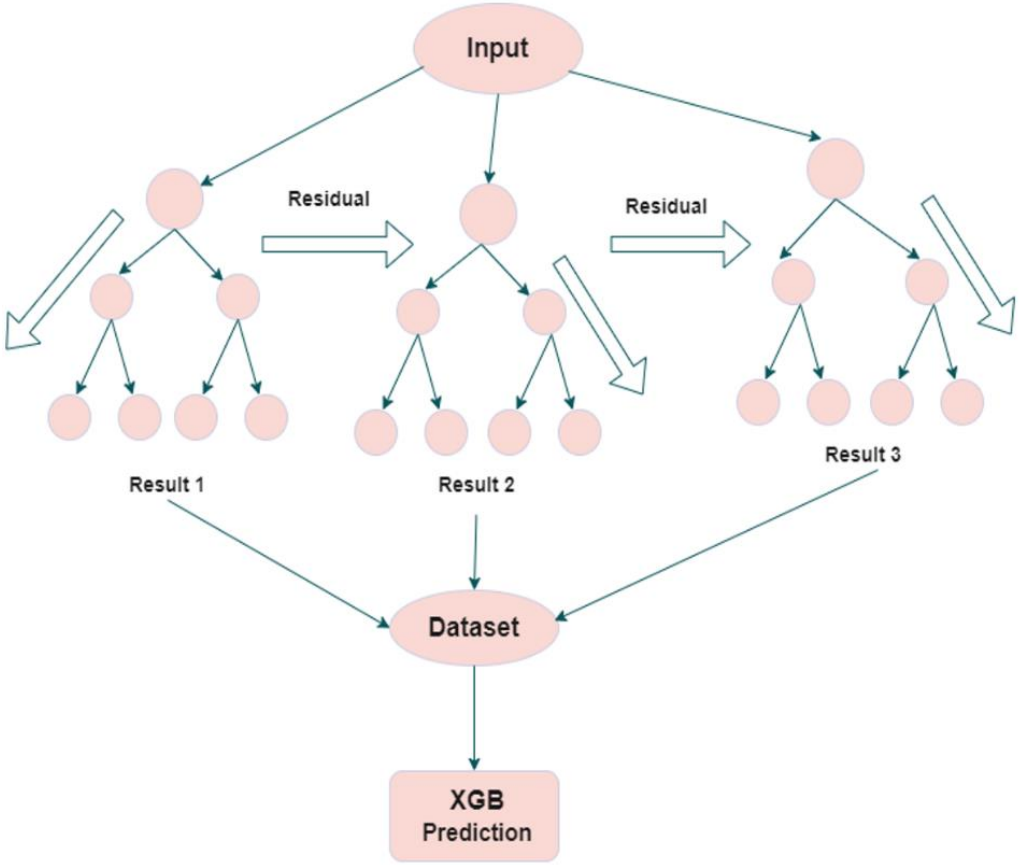

Figure 4.
XGBoost architecture.

3.4.1. XGBOOST

XGBoost is an advanced machine learning method that uses the gradient-boosting framework, an ensemble technique in which numerous weak learners, often decision trees, are combined to form a more robust prediction model. One of XGBoost's primary features is its use of regularization techniques like Lasso and Ridge, which help minimize overfitting by penalizing models that are too complex. XGBoost is known for its speed, accuracy, and versatility, making it ideal for various machine learning tasks such as classification, regression, and ranking, as shown in Figure 4.

Similarity Metrics

For Regression problem: $S_m = \frac{\sum(res)^2}{n_{res+\lambda}}$ (1)

Where λ = Hyperparameter.

For classification problem: $S_m = \frac{\sum(res)^2}{P_r+(1-P_r)}$ (2)

Where P_r = Probability of either left side or right side.

Information Gain:

Gain = (Sum of the similarity metrics of left and right sides) – (Similarity metric of the top branch) (3)

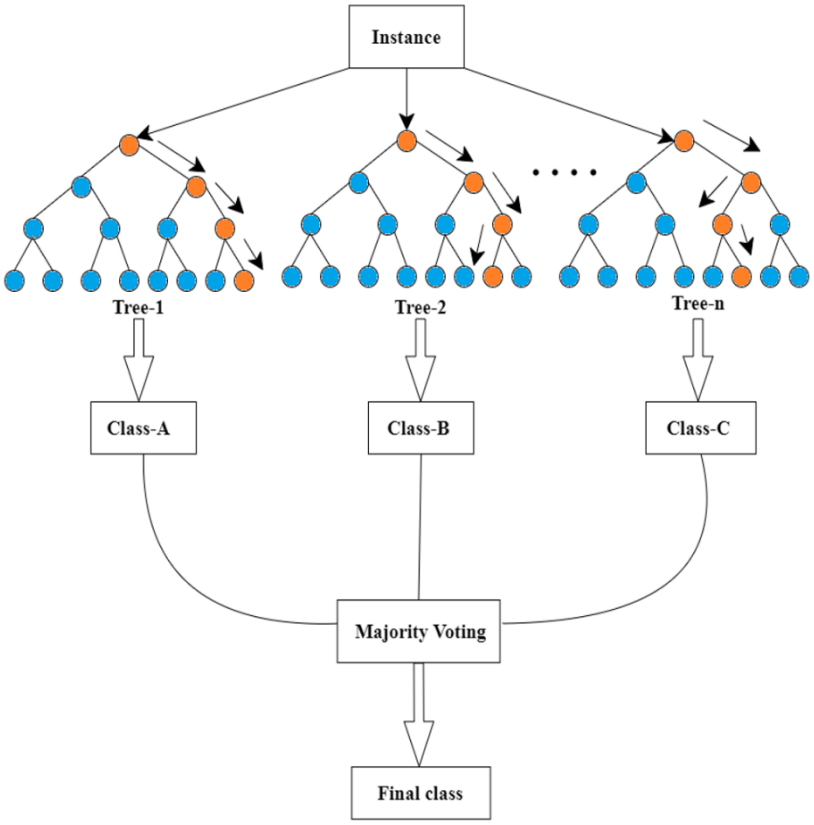

Figure 5.
Random forests architecture.

3.4.2. Random Forest

Leo Breiman and Adele Cutler developed Random Forest (RF), a prominent machine learning technique. It is primarily used for classification and regression problems. A Random Forest functions as a decision-making team, with multiple "trees" (individual decision trees) collaborating to improve forecast accuracy. In a Random Forest, each tree independently makes a prediction, and the final output is determined by aggregating these predictions—majority voting for classification tasks and averaging for regression tasks. This ensemble strategy reduces overfitting while increasing model resilience, as shown in Figure 5.

a) Gini Impurity (for Classification Trees): A randomly selected item's likelihood of being incorrectly categorized can be assessed using the Gini impurity. Accordingly, the Gini impurity indicates the level of informational disarray or unpredictability. The calculation of the Gini impurity (Gini(t)) for a node t in a binary classification problem with classes A and B is as follows:

$$Gini(t)=1-\sum iP(i|t)2 \quad (4)$$

Where P (i|t) is the probability of class i at node t.

a) **Entropy:** Decision trees and random forests frequently use entropy to determine splits.

Entropy (t) = $-\sum_{i=1}^{K} p(i|t)\log_2 p(i|t)$

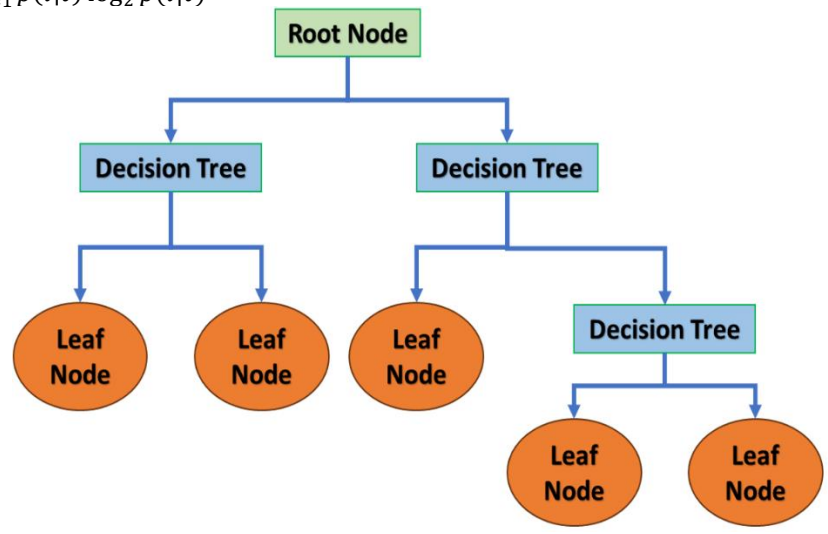

Figure 6.
Decision tree architecture.

3.4.3. Decision Tree

A decision tree is a supervised machine-learning approach that can be used for classification and regression tasks. It works by recursively dividing data into subsets based on the values of input attributes, creating a tree-like structure. In this structure, each internal node represents a feature, each branch represents a decision or condition based on that feature, and each leaf node contains the final prediction or result. The goal of a decision tree is to partition the data into subsets that are as homogeneous as possible, allowing for accurate and efficient predictions in classification and regression tasks, as shown in Figure 6.

a) Entropy: Entropy is a measure of disorder or unpredictability in a dataset. It is widely used in classification tasks to assess the uncertainty of class label distributions. It measures how mixed or pure a dataset is at a specific node in a decision tree. The mathematical definition of entropy (H_i) for a subset at the i[th] node with K classes is as follows:

$$H_i = \sum_{k \in K}^{n} p(i,k) \log_2 p(i,k) \quad (6)$$

Where,

- S represents the dataset sample.
- k denotes a specific class among K potential classes.
- $p\ (i,\ k)$ is the proportion of data points belonging to class k in the subset at node i, computed as follows:

$$p(k) = \frac{\text{Number of data points in class k}}{\text{Total number of data points in S}} \quad (7)$$

It is crucial that $p(i,k\ ) \neq 0$, as the logarithmic function is undefined for zero probability. A high entropy number suggests more disorder (more mixed classes), whereas a lower entropy indicates a purer node with a dominating class representation. This notion is critical in decision tree building since nodes are separated to minimize entropy and enhance classification accuracy.

b) Gini Impurity or Index: Gini Impurity, commonly known as the Gini Index, is a statistic used in decision tree algorithms to estimate a dataset's impurity or uncertainty. While it is helpful for binary and multiclass classification problems, it is most effective. Gini Impurity calculates the probability of mistakenly categorizing randomly selected data points if a label is allocated based on the dataset's class distribution.

$$\text{Gini Impurity} = 1 - \sum p_i^{\,2} \quad (8)$$

Here,

- p_i The proportion of elements in the set belongs to the i[th] category.

c) Information Gain: Information Gain is a statistic used in decision tree algorithms to measure the reduction in uncertainty or entropy when a dataset is divided based on a specific attribute. It determines how efficiently a particular characteristic splits the data into subsets that are more homogeneous regarding the goal variable. This metric is critical in directing feature selection during tree building, ensuring that the most informative features are chosen to increase the model's prediction accuracy.

Information gain is determined by the difference in entropy between the parent dataset and the weighted entropy of the child subsets following a split:

$$IG\ (S,\ A) = H(S) - \sum \frac{|S_i|}{|S|} H(S_i) \quad (9)$$

Where,

- H(S): Entropy of the entire dataset S.
- |S|: Total number of instances in the dataset.
- $|S_i|$ is the number of instances in the subset S_i with the value i for attribute A.
- $H(S_i)$: Entropy of the entire dataset S.

3.4.4. Stacking Model

Model stacking is an ensemble learning strategy that aggregates the outputs of numerous base models (or learners) into a higher-level model called the meta-model. Instead of depending on a single model for predictions, stacking combines various architectures and learning approaches to improve overall performance, as shown in Figure 7.

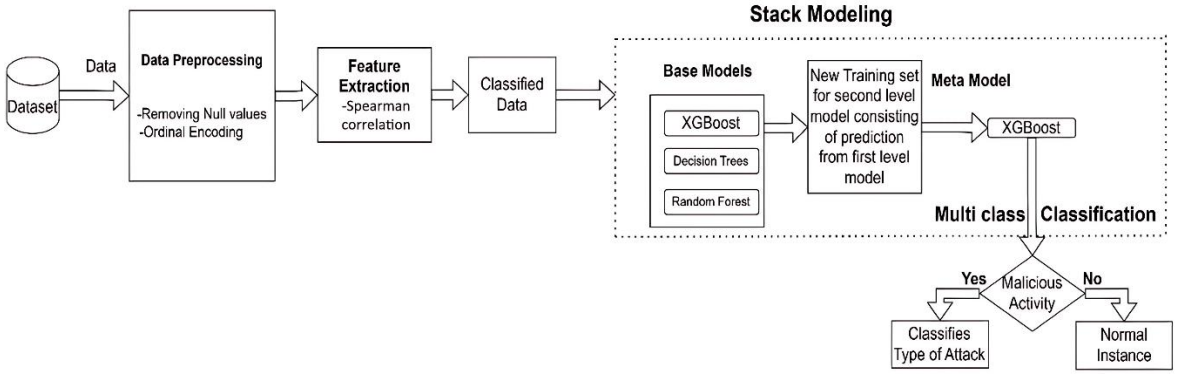

Figure 7.
Model stacking with original training features.

Working of Stacking Model:

1. Train Base Models: Train the base model using the training data.
2. Generate Predictions: Generating predictions from each base model using the validation data.
3. Create Meta-Features: Combine the predictions from the base models into meta-features.
4. Saving the XGBoost Model: The XGBoost model (XGB) is saved to a file named "xgboost_model.pkl" using the pickle.dump() function. This allows the trained model to be stored for later use.
5. Loading Models and Creating Stacking Classifier: The code imports the necessary libraries and loads the previously saved XGBoost, Random Forest, and Decision Tree models from their respective pickle files. A meta-learner is defined as an XGBoost classifier with specified hyperparameters. We create a stacking classifier (stacking_model) using the stacking class from scikit-learn.
6. Train Meta-Model: It uses the fit method to train the stacking model on the training data (X_train_1, y_train_encoded).
7. Make Predictions: We used the trained base and metamodels to predict new data.

4. Evaluation Metrics

4.1. Evaluation Metrics for Classification Models

Classification issues attempt to anticipate discrete outcomes, which are commonly represented as categories or classes. Numerous critical measures are used to evaluate the performance of these models, each of which provides a distinct view of how well a model works. The following is an in-depth summary of the most often-used metrics:

- Confusion Matrix: A tabular overview of actual versus projected classifications, the confusion matrix displays false positives (FP), false negatives (FN), true positives (TP), and true negatives (TN) in Table 1.

Table 1.
Confusion matrix.

	Predicted positive	**Predicted negative**
Actual positive	TP	FN
Actual negative	FP	TN

- Accuracy: Accuracy is one of the most understandable measures, expressing the proportion of correct forecasts to total guesses. It is computed as:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Toatal\ number\ of\ input\ samples} \quad (10)$$

While accuracy is helpful, it may not completely represent model performance in unbalanced datasets where one class dominates.

- Precision: The percentage of accurate positive projections is referred to as "precision." This indicator evaluates the accuracy of the model's positive predictions, making it relevant in situations where the cost of false positives is significant.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (11)$$

- Recall (or Sensitivity): The model's capacity to reliably recognize all positive cases is evaluated through the use of recall information. It is of utmost significance in situations when the cost of false negatives is substantial, such as in the field of medical diagnostics [40].

$$Recall = \frac{TP}{TP+FN} \quad (12)$$

- F1 Score: The F1 score is the harmonic average of precision and recall. It balances the trade-off between the two indicators, offering a more complete perspective on model performance.

$$\text{F1 Score} = 2.\frac{Precision\ .Recall}{Precision+Recall} \quad (13)$$

- ROC Curve: The curve known as the Receiver Operating Characteristic (ROC) displays the performance of the model in accordance with a number of different classification criteria. It plots:
- True Positive Rate (TPR) = Recall,
- False Positive Rate (FPR) is calculated as $FPR = \frac{FP}{FP+TN}$ (14)

An ideal model's ROC curve hugs the top-left corner, indicating high sensitivity and specificity.

Table 2.
Comparison table of evaluation metrics for standard models.

S. No	**Model**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 Score (%)**
1	XGBOOST	95.11	96.0	95.1	95.7
2	Decision tree	95.04	94.85	95.04	94.72
3	Random forest	94.08	94.12	94.08	93.38
4	Cat boost	92.17	91.03	92.17	90.46
5	Support vector machine	91.82	97.71	99.24	98.58
6	Logistic regression	78.40	76.25	78.25	75.43
7	Naïve bayes	66.09	68.50	66.09	63.74

4.2. Performance Analysis of Standard Models and Stacked Models

Table 2 compares the performance measures of several machine learning models.

- The first four models—XGBoost, Decision Tree, Random Forest, and CatBoost—provide higher accuracy, precision, recall, and F1 scores, suggesting their ability to handle the dataset effectively.
- Among them, XGBoost exceeds the rest in terms of accuracy (95.11%), precision (96.0%), recall (95.1%), and F1 score (95.7%), making it the most dependable model for the task.
- Naïve Bayes has the lowest accuracy (66.09%) and F1 score (63.74%), indicating its limitations in this dataset.
- Support Vector Machine (SVM) has remarkable precision and recall, but its overall accuracy (91.82%) is inferior compared to the top four models, potentially due to generalization or class imbalance trade-offs.

The top four models—XGBoost, Decision Tree, Random Forest, and CatBoost—were chosen for further examination based on their high performance in key parameters such as accuracy, precision, recall, and F1 score. XGBoost had superior accuracy and predictive capability, while Naïve Bayes had low accuracy and F1 score, making it unsuitable for this dataset. Table 3 examines the performance of various stacked model combinations, which combine base models (XGBoost, Decision Tree, Random Forest, and CatBoost) with a meta-model to improve prediction accuracy. Stacked models capitalize on the strengths of individual models, enhancing overall performance by correcting for their flaws. The best results were achieved by integrating XGBoost, Decision Tree, and Random Forest as base models, with XGBoost as the meta-model. This demonstrates the strength of ensemble learning.

Table 3.
Comparison table of evaluation metrics for stacked models.

S. No	Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
1	Base Models: XG, RF, DT, Meta, XG	96.37	96.39	96.37	96.25
2	Base Models: XG, DT, CB, Meta, RF	95.58	94.5	95.66	94.29
3	Base Models: XG, RF, CB, Meta, XG	95.55	94.44	95.56	94.1
4	Base Models: XG, RF, CB, Meta RF	95.52	94.41	95.58	94.16
5	Base Models: XG, DT, CB, Meta, XG	95.61	94.46	95.64	94.17
6	Base Models: XG, RF, DT, Meta RF	95.6	94.47	95.7	94.24
7	Base Models: RF, DT, CB, Meta, XG	94.92	94.07	94.9	93.9
8	Base Models: XG, RF, DT, Meta, CB	94.8	94.07	94.55	93.7
9	Base Models: XG, DT, CB, Meta, CB	94.87	94.08	94.63	93.59
10	Base Models: RF, DT, CB, Meta, CB	94.48	94.04	94.27	93.62
11	Base Models: RF, DT, CB, Meta, RF	94.81	94.27	94.75	94.01
12	Base Models; XG, RF, DT, Meta, DT	94.86	93.96	94.68	93.54
13	Base Models: XG, DT, CB, Meta, DT	94.53	93.42	94.34	93.04
14	Base Models; XG, RF, CB, Meta, CB	94.21	93.82	93.84	93.26
15	Base Models; XG, RF, CB, Meta, DT	94.64	93.54	94.49	93.16
16	Base Models; RF, DT, CB, Meta, DT	93.83	93.53	93.39	92.95

- Base Models: XG, RF, DT, Meta, XG (XGBoost), Random Forest, and Decision Tree as base models, with XGBoost as the meta-model) had the highest accuracy of 96.37%, as well as high precision (96.39%), recall (96.37%), and F1 score (96.25%). This demonstrates its superior predictive capacity and resilience when compared to other combinations.
- Other combinations, such as Base Models: XG, DT, CB, Meta, RF and Base Models: XG, RF, CB, Meta, XG, also performed well, although with lower accuracy and other assessment metrics.
- Lower-performing combinations, such as Base Models: RF, DT, CB, Meta, DT (with an accuracy of 93.83%), demonstrated lower performance, emphasizing the necessity of selecting appropriate base and meta-model combinations.

The combination of XGBoost, Random Forest, and Decision Tree as base models, with XGBoost as the meta-model, proved to be the most successful, achieving a 96.37% accuracy rate. This combination was chosen as the ideal stacked model for the challenge.

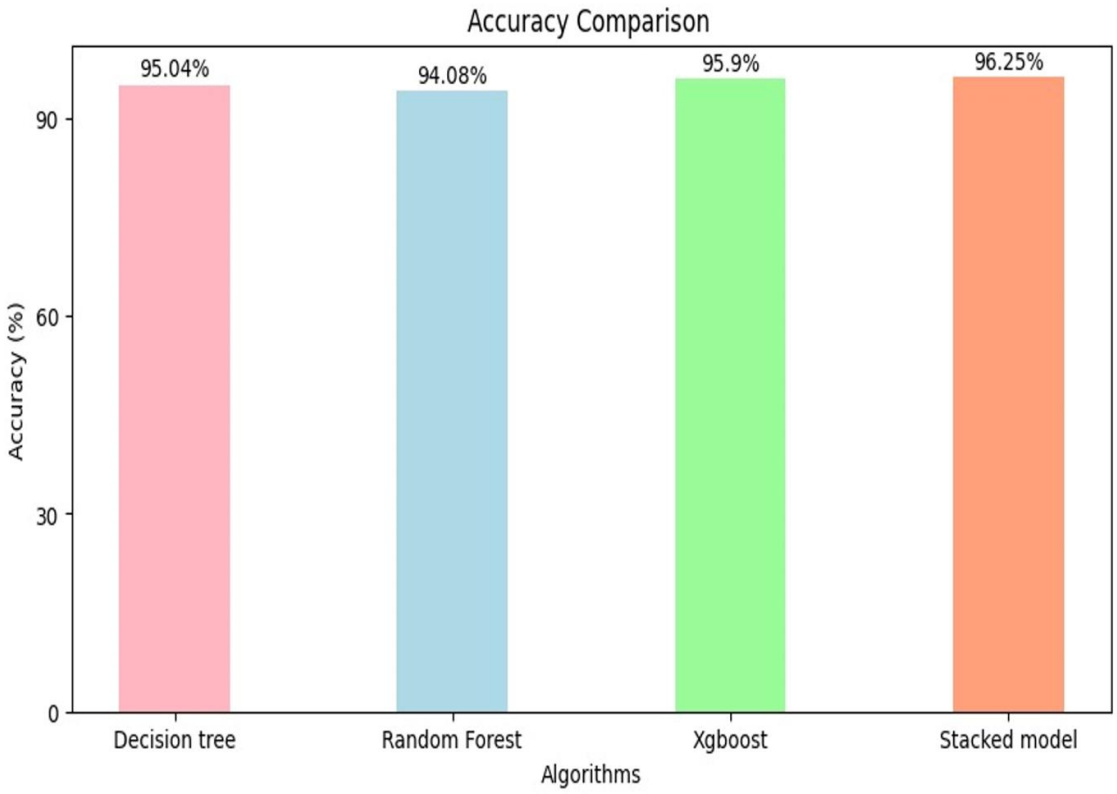

Figure 8.
Graph comparing the accuracy of individual standard models and meta-models.

Figure 8 illustrates the comparative accuracy of traditional models: Decision Tree (95.04%), Random Forest (94.08%), and XGBoost (95.9%). The stacked model, which integrates various algorithms, achieves the highest accuracy of 96.25%, showcasing its exceptional performance. This underscores the effectiveness of ensemble learning in enhancing predictive accuracy.

The comparison focuses on the following:

- XGBoost (Individual Model): Among the conventional models, XGBoost has the highest accuracy (95.11%), followed by Decision Tree (95.04%) and Random Forest (94.08%).
- Meta-model (Base Models: XG, RF, DT, Meta, XG): The stacked meta-model, which combines XGBoost, Random Forest, and Decision Tree as base models and XGBoost as meta-model, has the most excellent accuracy of 96.37%, beating all separate standard models.
- Ensemble Learning Impact: Stacking combines the strengths of numerous algorithms, resulting in higher prediction accuracy and robustness.

The study emphasizes the need to adopt ensemble techniques, such as stacking, to increase accuracy in complicated classification problems. Among the tested models:

- XGBoost emerged as the top individual model.
- The stacked meta-model (Base Models: XG RF DT Meta XG) has the highest overall accuracy, at 96.37%.
- These findings demonstrate the capacity of ensemble approaches to improve model dependability and predictive power under challenging datasets.

5. Conclusion

This paper focuses on the transformative potential of incorporating IoT, drones, and AI to enhance productivity, examining the integration of modern agricultural practices with cutting-edge technologies in Agriculture 4.0. Cybersecurity has become a very important issue because of these changes, and we need to take strong steps, such as using intrusion detection systems and machine learning solutions, to stop new threats. To guarantee the sustainability and prosperity of India's agricultural sector, it is imperative to promote collaborative endeavors between agricultural stakeholders and cybersecurity specialists. The vulnerabilities introduced by the convergence of technology and agriculture are addressed by this collaboration. Proactive measures and cross-sector partnerships are essential for strengthening resilience against evolving security challenges and fostering innovation. Acknowledging the risks associated with integrating technology in agriculture and establishing strategies to protect infrastructure security and data integrity is essential. To ensure the future security and prosperity of India's agricultural landscape, it is necessary to capitalize on the advantages of Agriculture 4.0 while simultaneously reducing its associated risks.

References

[1] Y. Liu, X. Y. Ma, L. Shu, G. P. Hancke, and A. M. Abu-Mahfouz, "From industry 4.0 to agriculture 4.0: Current status, enabling technologies, and research challenges," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4322–4334, 2021. https://10.1109/TII.2020.3003910

[2] M. A. Ferrag, L. Shu, O. Friha, and X. Yang, "Cyber security intrusion detection for agriculture 4.0: Machine learning-based solutions, datasets, and future directions," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 3, pp. 407–436, 2022. https://10.1109/JAS.2021.1004344

[3] G. Aceto, V. Persico, and A. Pescapé, "A survey on information and communication technologies for industry 4.0: State-of-the-art, taxonomies, perspectives, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3467-3501, 2019.

[4] Industry 4.0 and cybersecurity, "Industry 4.0 and cybersecurity: Managing risk in an age of connected production," Retrieved: https://www2.deloitte.com/content/dam/insights/us/articles/3749_Industry4-0_cybersecurity/DUP_Industry4-0_cybersecurity.pdf. [Accessed 2020.

[5] M. A. Ferrag, L. Shu, H. Djallel, and K. K. R. Choo, "Deep learning-based intrusion detection for distributed denial of service attack in agriculture 4.0," *Electronics*, vol. 10, no. 11, p. 1257, 2021. https://10.3390/electronics10111257

[6] K. Huang *et al.*, "Photovoltaic agricultural internet of things towards realizing the next generation of smart farming," *IEEE Access*, vol. 8, pp. 76300-76312, 2020. https://10.1109/ACCESS.2020.2988663

[7] O. Friha, M. A. Ferrag, L. Shu, and M. Nafa, "A robust security framework based on blockchain and SDN for fog computing enabled agricultural internet of things," in *In Proc. Int. Conf. Internet Things and Intelligent Applications, Zhenjiang, China, 2020, pp. 1−5*, 2020.

[8] W. J. Zhu, M. L. Deng, and Q. L. Zhou, "An intrusion detection algorithm for wireless networks based on ASDL," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 1, pp. 92–107, 2018. https://10.1109/JAS.2017.7510754

[9] M. Agarwal, S. Purwar, S. Biswas, and S. Nandi, "Intrusion detection system for PS-poll DoS attack in 802.11 networks using real time discrete event system," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 4, pp. 792–808, 2017. https://10.1109/JAS.2016.7510178

[10] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, p. 102419, 2020.

[11] R. A. Disha and S. Waheed, "Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique," *Cybersecurity*, vol. 5, no. 1, p. 1, 2022. https://10.1186/S42400-021-00103-8/TABLES/10

[12] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1-22, 2019. https://doi.org/10.1186/s42400-019-0038-7

[13] I. Dutt, S. Borah, and I. K. Maitra, "Immune system based intrusion detection system (IS-IDS): A proposed model," *IEEE Access*, vol. 8, pp. 34929-34941, 2020. https://10.1109/ACCESS.2020.2973608

[14] J. R. Rose, M. Swann, G. Bendiab, S. Shiaeles, and N. Kolokotronis, "Intrusion detection using network traffic profiling and machine learningfor IoT," presented at the IEEE 7th International Conference on Network Softwarization (NetSoft), Tokyo, Japan, 2021, pp. 409-415, doi: https://10.1109/NetSoft51509.2021.9492685, 2021.

[15] A. Thakkar and R. Lohiya, "A review of the advancement in intrusion detection datasets," *Procedia Computer Science*, vol. 167, pp. 636-645, 2020. https://doi.org/10.1016/j.procs.2020.03.330

[16] M. Amar and E. O. Bouabid, "Hybrid intrusion detection system using machine learning," *Network Security*, vol. 2020, no. 5, pp. 8-19, 2020. https://doi.org/10.1016/S1353-4858(20)30056-8

[17] G. D. C. Bertoli *et al.*, "An end-to-end framework for machine learning-based network intrusion detection system," *IEEE Access*, vol. 9, pp. 106790-106805, 2021. https://10.1109/ACCESS.2021.3101188

[18] R. Magán-Carrión, D. Urda, I. Díaz-Cano, and B. Dorronsoro, "Towards a reliable comparison and evaluation of network intrusion detection systems based on machine learning approaches," *Applied Sciences*, vol. 10, no. 5, p. 1775, 2020. https://doi.org/10.3390/app10051775

[19] D. Urda, I. Díaz-Cano, and B. Dorronsoro, "Towards a reliable comparison and evaluation of network intrusion detection systems based on machine learning approaches," *Applied Sciences*, vol. 10, p. 1775, 2020. https://doi.org/10.3390/app10051775

[20] M. Prasad, S. Tripathi, and K. Dahal, "An efficient feature selection based Bayesian and Rough set approach for intrusion detection," *Applied Soft Computing*, vol. 87, p. 105980, 2020. https://10.1016/J.ASOC.2019.105980

[21] A. A. Megantara and T. Ahmad, "A hybrid machine learning method for increasing the performance of network intrusion detection systems," *Journal of Big Data*, vol. 8, no. 1, p. 142, 2021. https://doi.org/10.1186/s40537-021-00531-w

[22] M. Wang, K. Zheng, Y. Yang, and X. Wang, "An explainable machine learning framework for intrusion detection systems," *IEEE Access*, vol. 8, pp. 73127-73141, 2020. https://10.1109/ACCESS.2020.2988359

[23] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. A. Khan, "Performance analysis of machine learning algorithms in intrusion detection system: A review," *Procedia Computer Science*, vol. 171, pp. 1251-1260, 2020. https://10.1016/j.procs.2020.04.133

[24] B. Soumya and S. Kapil, "Intrusion detection system in IoT network using ML," *NeuroQuantology*, vol. 20, no. 13, pp. 3597-3601, 2022. https://10.14704/nq.2022.20.13.NQ88441

[25] F. Nabi and X. Zhou, "Enhancing intrusion detection systems through dimensionality reduction: A comparative study of machine learning techniques for cyber security," *Cyber Security and Applications*, p. 100033, 2024. https://doi.org/10.1016/j.csa.2023.100033

[26] A. H. Azizan *et al.*, "A machine learning approach for improving the performance of network intrusion detection systems," *Annals of Emerging Technologies in Computing*, vol. 5, no. 5, pp. 201-208, 2021. https://10.33166/AETiC.2021.05.025

[27] T. H. Aldhyani and H. Alkahtani, "Cyber security for detecting distributed denial of service attacks in agriculture 4.0: Deep learning model," *Mathematics*, vol. 11, no. 1, p. 233, 2023. https://doi.org/10.3390/math11010233

[28] S. Ho, S. Al Jufout, K. Dajani, and M. Mozumdar, "A novel intrusion detection model for detecting known and innovative cyberattacks using convolutional neural network," *IEEE Open Journal of the Computer Society*, vol. 2, pp. 14-25, 2021. https://10.1109/OJCS.2021.3050917

[29] P. Sun *et al.*, "DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system," *Security and communication networks*, vol. 2020, no. 1, p. 8890306, 2020. https://doi.org/10.1155/2020/8890306

[30] M. Di Mauro, G. Galatro, and A. Liotta, "Experimental review of neural-based approaches for network intrusion management," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2480-2495, 2020. https://10.1109/TNSM.2020.3024225

[31] I. Hidayat, M. Z. Ali, and A. Arshad, "Machine learning-based intrusion detection system: An experimental comparison," *Journal of Computational and Cognitive Engineering*, vol. 2, no. 2, pp. 88-97, 2023. https://10.47852/bonviewJCCE2202270

[32] S. M. Kasongo, "A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework," *Computer Communications*, vol. 199, pp. 113-125, 2023. https://doi.org/10.1016/j.comcom.2022.12.010

[33] M. A. Khan, N. Iqbal, H. Jamil, and D.-H. Kim, "An optimized ensemble prediction model using AutoML based on soft voting classifier for network intrusion detection," *Journal of Network and Computer Applications*, vol. 212, p. 103560, 2023. https://10.1016/J.JNCA.2022.103560

[34] A. O. Alzahrani and M. J. Alenazi, "Designing a network intrusion detection system based on machine learning for software defined networks," *Future Internet*, vol. 13, no. 5, p. 111, 2021. https://doi.org/10.3390/fi13050111

[35] R. El-Sayed, A. El-Ghamry, T. Gaber, and A. E. Hassanien, "Zero-day malware classification using deep features with support vector machines," in *2021 Tenth International Conference on Intelligent Computing and Information Systems (ICICIS)*, 2021: IEEE, pp. 311-317, doi: https://10.1109/ICICIS52592.2021.9694256.

[36] V. Priyanka and T. Gireesh Kumar, "Performance assessment of IDS based on CICIDS-2017 dataset," in *Information and Communication Technology for Competitive Strategies (ICTCS 2020) ICT: Applications and Social Interfaces*, 2022: Springer, pp. 611-621.

[37] N. Thapa, Z. Liu, D. B. Kc, B. Gokaraju, and K. Roy, "Comparison of machine learning and deep learning models for network intrusion detection systems," *Future Internet*, vol. 12, no. 10, p. 167, 2020. https://10.3390/fi12100167

[38] H. Saleh, A. Alharbi, and S. H. Alsamhi, "OPCNN-FAKE: Optimized convolutional neural network for fake news detection," *IEEE Access*, vol. 9, pp. 129471-129489, 2021. https://10.1109/ACCESS.2021.3112806

[39] F. Kha, "Advancing machine learning: Development, evaluation, and feature engineering in domain-specific applications," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 12, no. 2, pp. 415–423, 2024.

[40] A. Bhardwaj, R. Tyagi, N. Sharma, A. Khare, M. S. Punia, and V. K. Garg, "Network intrusion detection in software defined networking with self-organized constraint-based intelligent learning framework," *Measurement: Sensors*, vol. 24, p. 100580, 2022. https://10.1016/J.MEASEN.2022.100580