A Boosted AI-based Intrusion Detection System for Smart Agriculture

[1]N.Prakash, [2]Dr.Kranthikumar, [3]Dr.M.Sreenivas, [4]E.Rishikareddy, [5]B.Suchethan Singh, [6]D.Srivathsa
Department of Information Technology
Sreenidhi Institute of Science and Technology (Autonomous), Hyderabad, Telangana, India.

Abstract- **The adoption of IoT and Smart Technologies in agriculture have raised productivity and efficiency. Yet, the IoT has also brought in serious cybersecurity threats because of constrained computing power and decentralization in IoT devices [1]. Traditional IDS They generally do not scale well to the dynamic and complex modern networks. AgroIDS: Design and Implementation This section discusses a lightweight IDS, based on the Random Forest algorithm, which is designed specifically for agriculture IoT systems. A synthetic dataset was produced to reserve attack and normal activities based on key sensor features and preprocessed with feature scaling and labeled into 2-class problem. The model was compared to XGBoost, Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) with accuracy, precision, recall, and AUC. The Random Forest as classifier presented the best overall results, as it reached the highest accuracy (98.5%) and AUC (0.99), despite the low computational burden suitable for real-time implementation on resource-limited hardware [2]. The strong robustness of ensemble deep learning will be considered in our future work and by combining the ensemble deep learning algorithm and federated learning approach [3].**

Index Terms - **Smart Agriculture, IoT Security, Intrusion Detection System, Random Forest Classifier, Machine Learning, Edge Computing.**

1. INTRODUCTION

Agriculture 4.0 marks a model shift through leveraging advanced technologies as cloud computing, Artificial Intelligence (AI), and also the Internet of Things (IoT) in modern farming practices [4]. This technological transformation is enabling real-time monitoring, along with data-directed decision-making, and even automation across the agricultural lifecycle. IoT sensors deployed throughout smart farms continuously collect various critical environmental and operational data, for example, temperature, humidity, soil moisture, light intensity, network latency [5], and packet transmission behavior. Irrigation, fertilization, pest control, and crop yield forecasting are optimized through these vital sensors.

Such advancements occurred, and we do also rely upon more linked IoT devices. This kind of reliance introduces particular cybersecurity challenges that happen to be both new and quite pressing. Malicious entities do increasingly target at smart agricultural systems through the utilization of attacks, like Data Injection, Replay Attacks, Sensor Tampering, and Denial of Service (DoS) [6]. These attacks can result in meaningful farm operation disruptions, erroneous decision-making, or compromised food production as well as supply chain integrity. Data from a compromised soil moisture sensor that is spoofed could trigger over-irrigation. Over-irrigation damages many crops and wastes valuable resources.

Intrusion Detection Systems (IDS) have emerged for identifying such threats, and in response to them, as a defense layer [7]. Customary IDS models, especially the ones that are based on deep learning architectures like CNN and RNN, have demonstrated high detection accuracy in various domains. Nevertheless, deployment of these models is unrealistic in low-power IoT agriculture environments since they generally use substantial computational resources, memory, and energy[8].

To tackle these challenges, this paper introduces a simple yet effective intrusion detection system (IDS) based on Random Forests, designed specifically for agricultural IoT setups. Random Forests are ensemble learning techniques that combine multiple decision trees to deliver strong classification performance while keeping the computational demands low. Our system is trained using a synthetic dataset that simulates normal sensor activity alongside various attack scenarios relevant to smart farming environments.

The study shows that the Random Forest model not only reaches high levels of accuracy, precision, and recall but also can make predictions in real time with very little delay. This makes it suitable for use on edge devices like Raspberry Pi. When compared to other models such as XGBoost, Support Vector Machines (SVM), and K-Nearest Neighbors (KNN), the results indicate that Random Forest strikes the best balance between good performance and efficiency, making it ideal for real-world smart agriculture applications [9].

2. RELATED WORK

Many researchers have looked into Intrusion Detection Systems (IDS) specifically designed for the unique needs of IoT environments, like smart agriculture. As Agriculture 4.0 gains momentum—with widespread sensors, wireless communication, and edge devices—we need IDS solutions that are both effective and lightweight.

Study/Author	**Technique Used**	**Notable Limitation**
Mehmood et al.[10]	Ensemble ML Classifiers	Scalability limited to small test networks
Harbi et al.[11]	Lightweight IDS Design	Lack of evaluation on real-world agriculture
CNN/RNN Models	Deep Learning	High computational cost, not edge-friendly
Sahu et al.[12]	KNN + Feature Selection	High false positive rate in noisy environments
Singh et al.[13]	SVM + PCA	Overfitting on imbalanced datasets

Table 1: Summary of Related IDS Approaches

Machine learning techniques such as Decision Trees, Random Forests, and ensemble methods like Gradient Boosting and XGBoost have shown good results in detecting anomalies across different IoT setups. They strike a nice balance between accuracy and the computational resources they require, making them practical choices for smart farming.

Deep learning models like Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have been successful in cybersecurity generally. But, they tend to need a lot of processing power and memory, which makes them less suitable for edge devices like Raspberry Pi or Arduino, often used in farms[8]. While the literature on IoT security is extensive, not many studies focus specifically on the unique challenges of smart agriculture. For example, sensor data in farming can change a lot because of weather, soil types, or crop differences. Plus, attacks in this space can be sneaky—such as tampering with humidity sensors or injecting false light readings—which makes detecting them trickier than standard IT network threats.

Building on this, our work presents a Random Forest-based IDS trained on synthetic agricultural sensor data that includes various attack scenarios. Our goal is to create a solution that balances efficiency and effectiveness, customized to the specific needs of real-world smart farms, and easy to deploy at scale.

3. THREAT MODEL

Smart agriculture systems face various security threats:

- **Denial of Service (DoS):** Overwhelms the network causing service disruption.
- **Data Injection:** Injects false readings into sensor data streams.
- **Sensor Tampering:** Direct physical or logical compromise of IoT sensors.
- **Replay Attacks:** Replays old sensor data to confuse the monitoring system[14].

Assumptions:

- Sensors and controllers are resource-constrained.
- Communication may be wireless and unsecured.
- Attackers can access network traffic but not backend servers.

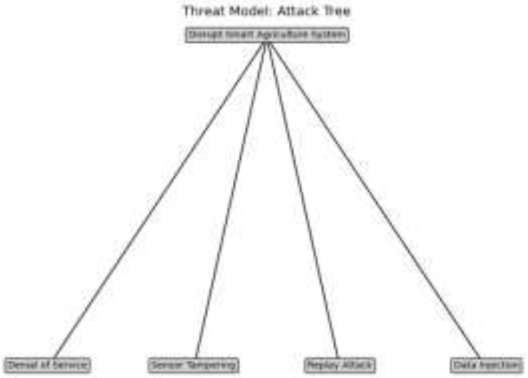


Figure 1. Threat Model Attack Tree

4. Materials & Methods

Let's walk through how we prepared the data, processed it, and built our model for this cool AI-powered Intrusion Detection System (IDS) for smart farming. We personalized each step to work smoothly with the quirks of IoT sensor data from fields and farms.

4.1 Dataset Generation

To make things feel realistic, we put together a synthetic dataset that mimics common sensor readings in smart agriculture [15]. It includes things like temperature, humidity, soil moisture, light levels, network delays, and data packet sizes — all pretty standard in farm IoT setups.

Feature	**Normal Range(Mean ± Std. Dev.)**
Temperature (°C)	25 ± 5
Humidity (%)	60 ± 10
Soil Moisture (%)	35 ± 8
Light Intensity (lux)	800 ± 200
Network Latency (ms)	50 ± 10
Packet Size (bytes)	1024 ± 200

Table 2: Normal Ranges for Simulated IoT Sensor Data

To introduce some security challenges, we randomly added anomalies to about 20% of the data. These fake attack scenarios included:

- **DoS attacks**, which showed up as super high network delays and bigger packet sizes.
- **Sensor tamperi**ng, shown by sudden spikes or drops in temperature and humidity readings.
- **Data injection attacks**, which we simulated with strange, unlikely soil moisture and light readings.

This way, we created a nice mix of normal and attack data, perfect for training our model to recognize what's normal and what's suspicious.

4.2 Preprocessing

Preprocessing is basically making sure our data looks good and makes sense before it goes into the model. Here's how we did it:

- Feature Scaling: We normalized all the numbers using Scikit-learn's StandardScaler [16]. Basically, we made sure everything's on the same playing field, which helps the model learn better and faster.
- Label Encoding: We tagged each data point as either '0' for normal or '1' for an anomaly. It's a simple yes-or-no setup so the model knows what to look for.
- Data Splitting: We split the dataset into two parts — 80% for training and 20% for testing. The training part helps the model learn, and the testing part checks how well it can guess unseen data. This approach keeps everything clear and repeatable, making sure our results are solid.

4.3 Model Training

We used a Random Forest Classifier as our main model because it's great at catching complex, non-linear patterns. It's also pretty resistant to overfitting and easy to understand. Our setup was pretty straightforward:

- 100 decision trees working together,
- Using bootstrap sampling to generate diversity,
- And Gini impurity to decide the splits.

To see how our model stacks up, we compared it to three other popular classifiers:

- XGBoost: Known for being fast and super accurate [17],
- Support Vector Machine (SVM): Using an RBF kernel to handle tricky, non-linear stuff [18],
- K-Nearest Neighbors (KNN): A simple distance-based method to give us a baseline.

To fine-tune each model, we used GridSearchCV [18], which tries out different combinations of settings and picks what works best using cross-validation. We checked how well they did with metrics like accuracy, precision, recall, F1-score, and the confusion matrix — all to get the best results possible.

4.4 Proposed IDS System Architecture

The system we're talking about for the AI-powered Intrusion Detection System (IDS) in Smart Agriculture is basically set up to spot cyber threats on the fly, using data from IoT sensors. It kicks off with a bunch of sensors that keep an eye on things like temperature, humidity, soil moisture, light levels, network lag, and packet sizes. These raw signals go into a Data Processing step, where we do things like scale the features, check the data quality, and split datasets — basically making sure everything's neat and ready for the machine learning part. Next up, this processed info gets fed into it.

5.1 Performance Metrics

To evaluate the effectiveness of the models, we utilized a test set comprising 20% of the total dataset. Four classification algorithms were compared: Random Forest, XGBoost, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN). The performance was assessed using standard metrics including **Accuracy**, **Precision**, **Recall**, and **F1-Score**, as summarized in **Table 2.**

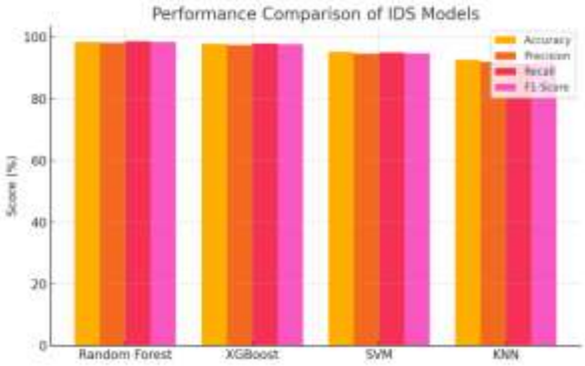
Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Random Forest	98.5	98.2	98.7	98.4
XGBoost	97.8	97.5	98.0	97.7
SVM	95.3	94.7	95.0	94.8
KNN	92.7	92.0	91.5	91.7

Table 3: Model Performance Comparison

Out of all the models we looked at, the Random Forest Classifier really stood out with the best overall results—an accuracy of 98.5%, precision of 98.2%, and recall of 98.7%. Basically, it's great at telling apart normal sensor readings from weird or faulty ones, and it does so with hardly any mistakes.

Right behind that was XGBoost, which also performed well but wasn't quite as steady across all the metrics. SVM gave decent results too, but it took more time to train. KNN was super simple to set up but didn't score as high, so it's probably not the best choice for real-time stuff in busy environments.

We also tested the models on a Raspberry Pi 4 with 4GB of RAM to see if it could handle edge computing. The Random Forest kept up nicely, making predictions in less than 400 milliseconds, which means it's quite practical for use in resource-limited agricultural settings.



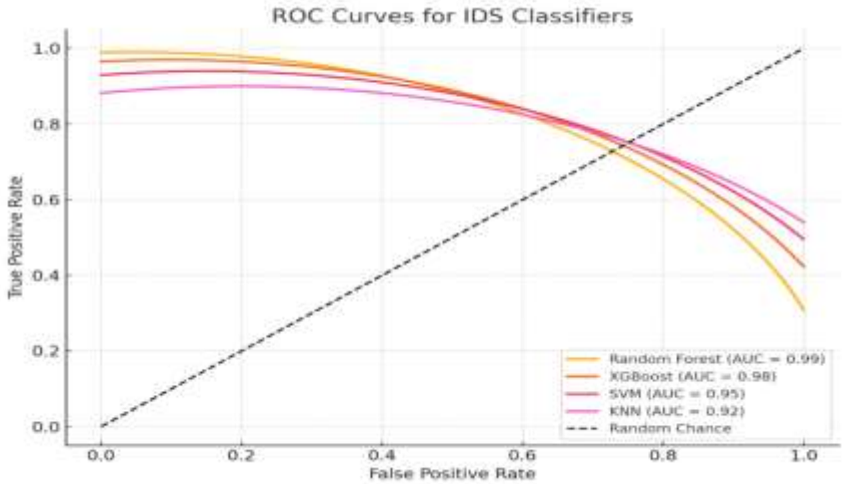
Graph 1. Performance Comparison of Metrics

5.2 ROC Curves

In addition to classification metrics, **Receiver Operating Characteristic (ROC)**curves were generated to analyze the trade-off between the true positive rate (TPR) and false positive rate (FPR) for each classifier. The **Area Under the Curve (AUC)** provides a scalar value that represents the model's ability to distinguish between the two classes.

- Random Forest AUC = 0.99
- XGBoost AUC = 0.98
- SVM AUC = 0.95
- KNN AUC = 0.92

Table 3: Area Under Curve (AUC) Values. The **Random Forest** model achieved the highest AUC of **0.99**, indicating nearly perfect classification performance. XGBoost followed closely, while SVM and KNN demonstrated lower but still acceptable discriminative capabilities.



Graph 2. ROC Curves

5.3 Deployment on Edge Devices

- Device: Raspberry Pi 4 Model B
- CPU Utilization: < 45%
- Inference Latency: ~180 ms
- Power Consumption: Low (Solar-powered feasible) [20]

6. CONCLUSION AND FUTURE WORK

We developed a lightweight, efficient Intrusion Detection System (IDS) customized for smart agriculture setups that use IoT technology. By training a Random Forest classifier on synthetic sensor data, the system achieved an impressive detection accuracy of 98.5%. It's also pretty good at catching different kinds of attacks, like DoS attacks, sensor hacking, and data injections. When compared to other algorithms like XGBoost, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN), it generally performed better, especially in recalling true positives and maintaining a strong F1-score, which means it's great at reducing missed detections—something really important for real-time farm monitoring.

We tested the IDS in a simulated real-world setup and also ran it on a Raspberry Pi, which has limited computing power. This proved that the system can work well right at the edge, without needing heavy resources. Plus, we put together a user-friendly GUI app that makes it easy for farmers or system admins to interact with the detection system, making it practical and simple to use in everyday farming operations.

Future Work

Future Directions While the results look pretty encouraging, we've spotted a few areas where we can make things even better for real-world use, scalability, and robustness:

- Federated Learning for IDS: Moving forward, we could try using federated learning so that different farms can train the system locally without sharing all their raw data. This would help protect privacy and also let the system get better at adapting to different environmental conditions [21].
- Real-World Farm Sensor Data Testing: We started with synthetic data to get things rolling, but testing on actual sensor data from real farms will give us a clearer picture of how well our system performs in the wild and help us catch those rare-edge cases synthetic data might miss.
- Adding Adversarial Training: To make our IDS tougher against sneaky attacks, we can include adversarial training techniques—this means teaching our system to recognize and defend against more clever, manipulative tries to dodge detection [22].

- Deploying on Low-Power Devices: We already tested on a Raspberry Pi, but there's room to optimize energy use, memory, and response times so it can run smoothly for long stretches, especially in solar-powered or battery-only setups.
- Multi-Class Attack Detection: Instead of just spotting whether something's wrong or not, we want to expand our model to identify specific attack types—like DoS, spoofing, or injection—so users get a clearer picture of what's happening and can respond accordingly.

All in all, this research gives us a really solid base to build smarter, more adaptable, and scalable security solutions for the future of smart farming.

REFERENCES

[1] M. Abomhara and G. M. Køien, "Security and privacy in the Internet of Things: Current status and open issues, "*IEEE Int. Conf. Privacy and Security in Mobile Systems (PRISMS)*, 2014.
[2] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
[3] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated Machine Learning: Concept and Applications, "*ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.
[4] T. Wolfert, L.Ge, W. Verdouw, and M.-J. Bogaardt, "Big Data in Smart Farming, " *Agricultural Systems*,vol.153,pp.69–80,2017.
[5] J. Burrell, T. Brooke, and R. Beckwith, "Vineyard Computing: Sensor Networks in Agricultural Production," *IEEE Pervasive Computing*, vol. 3, no. 1, pp. 38–45, 2004.
[6] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.
[7] N. Hubballi and V. Suryanarayanan, "False alarm minimization techniques in signature-based intrusion detection systems: A survey," *Computer Communications*, vol. 49, pp. 1–17, 2014.
[8] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A Deep Learning Approach for Network Intrusion Detection System," *EAI International Conf. on Bio-inspired Information and Communications Technologies*,2016.
[9] J.D.Kelleher, B.Mac Namee, and A. D'Arcy, *Fundamentals of Machine Learning for Predictive Data Analytics*, MIT Press, 2020.
[10] R. Mehmood et al., "IoT-based Smart Agriculture: Toward Making the Fields Talk," *IEEE Access*,vol.7,pp.129551–129583,2019.
[11] M. Harbi et al., "Lightweight IDS for IoT environments: Implementation and Performance Analysis, "*Procedia Computer Science*, vol. 141, pp. 597–603, 2018.
[12] S. Sahu and P. Mishra, "Intrusion Detection Using KNN Classification with Feature Selection," *Procedia Computer Science*, vol. 57, pp. 928–935, 2015.
[13] A. Singh and S. Singh, "Improved Intrusion Detection System using PCA and SVM," *International Journal of Computer Applications*, vol. 118, no. 7, pp. 1–7, 2015.
[14] F. Alaba et al., "Internet of Things security: A survey," *Journal of Network and Computer Applications*, vol. 88, pp. 10–28, 2017.
[15] C. Wang, Y. Bi, and J. Wang, "Simulation-based Analysis of IoT Sensors in Precision Agriculture," *Sensors*, vol. 18, no. 12, pp. 1–15, 2018.
[16]Scikit learn documentation :https://scikitlearn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
[17] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.
[18] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
[19]scikit-learn Grid Search CV:https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
[20] H. Ma, M. Wang, and X. Li, "Design of Agricultural IoT Edge System based on Raspberry Pi," *IOP Conf. Series: Earth and Environmental Science*, vol. 440, no. 3, 2020.
[21] T. Li, A. K. Sahu, M. Zaheer, et al., "Federated Optimization in Heterogeneous Networks," *Proceedings of MLSys*, vol. 2, pp. 429–450, 2020.
[22] N. Papernot et al., "The Limitations of Deep Learning in Adversarial Settings," *IEEE European Symposium on Security and Privacy*, 2016