Machine Learning-Based Intrusion Detection and Prediction in Wireless Sensor Networks for Enhanced Cybersecurity

Sachin Chavalagi
Department of E&C Engineering
BMSCE
Bengaluru, India.
Sachinchavalagi.ldc23@bmsce.ac.in

Abstract— Wireless Sensor Networks (WSNs) play a crucial role in a wide range of applications, such as industrial automation, smart cities, and healthcare. However, they are susceptible to cyber threats due their distributed and resource-constrained nature. In order to improve WSN cybersecurity, this study proposes an intrusion detection and prediction system based on machine learning (ML). To spot irregularities and anticipate possible security breaches in real time, the suggested solution makes use of both supervised and unsupervised machine learning Techniques for feature extraction and selection are used to increase detection accuracy and decrease the complexity of the data. To merge several models for reliable performance, the system also uses ensemble learning. Model training and evaluation are conducted using an extensive dataset including a range of a attack types attack types, including Sybil, sinkhole, and denial-of-service (DoS) attacks. The suggested ML-based method delivers excellent Detection accuracy with low false positives rates, and quick response times, according to experimental results. This solution ensures dependable and secure network operations by strengthening WSN security and offering predictive insights to stop future breaches.

1. INTRODUCTION

The foundation of contemporary industrial transformation is the idea of Industry 4.0, which combines digital and physical processes to create intelligent, networked, and adaptable ecosystems. The smooth integration of cyber-physical systems (CPS), cloud computing, The Internet of Things (IoT), and artificial intelligence (AI) is what sets apart Industry 4.0 from The earlier industrial revolutions were defined by mechanical, electrical, and digital automation [1], [2]. Real-time data collection, predictive decision-making, and autonomous operations are made possible by this integration, which benefits factories and industrial systems and maximizes resilience, cost savings, and productivity [3]. Industry 4.0 creates an environment where data-driven intelligence propels competitiveness Across industries including manufacturing and healthcare, agriculture, and logistics by connecting machines, devices, and people via sophisticated networks [4], [5].

WSN security a big problem even these advantages. They are susceptible to different types of cyberattacks because of their intrinsic features, such as distributed architecture, wireless connection, and constrained processing resources [11]. Frequent dangers include flooding attacks, which overload the network with too much traffic; scheduling attacks, which alter timing mechanisms to cause collisions or delays; blackhole attacks, in which malevolent nodes drop packets to obstruct communication; and Grayhole attacks, in which selective dropping compromises particular data flows [12], [13]. Data loss, system outages, and even safety risks in industrial settings might result from such attacks. Furthermore, these vulnerabilities are exacerbated by Industry 4.0's growing interconnectedness, since a single hacked sensor node might have a domino effect on the entire system [14].

Recent industrial statistics demonstrate the importance of this issue. According to monthly reports of penetration attempts that interrupt production lines and supply chains globally, manufacturing has continuously been one of the most targeted areas for cyberattacks [15]. These attacks have repercussions that go beyond monetary losses; they may jeopardize worker safety, product quality, and operational confidence. Because of this, cybersecurity is not an add-on feature but rather a key enabler of Industry 4.0. However, because of their high false alarm rates, lack of adaptation, and incapacity to identify new attack patterns, traditional Intrusion detection and prevention systems (IDPS) frequently fail in WSN situations [16]. T The necessity to create intelligent, predictive frameworks that can effectively monitor, identify, and mitigate intrusions in real time is highlighted by this constraint.

As a result, Researchers are increasingly utilizing machine learning. (ML) and deep learning (DL) techniques, which offer strong instruments for proactive threat analysis, attack classification, and anomaly identification [17]. For instance, Multilayer Perceptron's (MLP) are good at capturing intricate nonlinear correlations in incursion datasets, whereas Decision Tree (DT) classifiers provide interpretable structures to distinguish between normal and aberrant traffic patterns [18]. By learning compressed representations of typical behaviour and identifying deviations, Autoencoders (AE), which are unsupervised models, are also well-suited for anomaly detection [19]. When used for intrusion detection in WSNs Studies have shown that these models perform better in terms of accuracy, recall, and detection speed than traditional methods like Random Forests (RF) or Logistic Regression (LR) [20].

Modern frameworks stress the need of situation-based prioritization processes In addition to classification accuracy. Prioritization guarantees that Identified risks are ranked according to their seriousness, operational impact, and application context, in contrast to traditional systems that handle all warnings equally [21]. For example, a Grayhole incursion in environmental monitoring might not be as serious as a blackhole attack in a robotics control system. Resources can be distributed more efficiently by intelligently ranking threats, guaranteeing that high-risk breaches are promptly neutralized while reducing needless overhead for less serious situations [22]. This move to context-aware intrusion detection is in keeping with Industry 4.0's overarching objectives, which place a premium on responsiveness and dependability.

Thus, a situation-aware predictive framework for intrusion

detection and prevention in Industry 4.0 is established by combining ML/DL techniques with intelligent prioritizing. Predictive techniques proactively examine both historical data and real-time streams to foresee weaknesses and put countermeasures in place before disruptions happen, in contrast to reactive tactics, which act only after an attack has been detected [23]. By lowering downtime, guaranteeing business continuity, protecting sensitive industrial data, and enhancing detection performance, these frameworks increase the resilience of industrial WSNs.

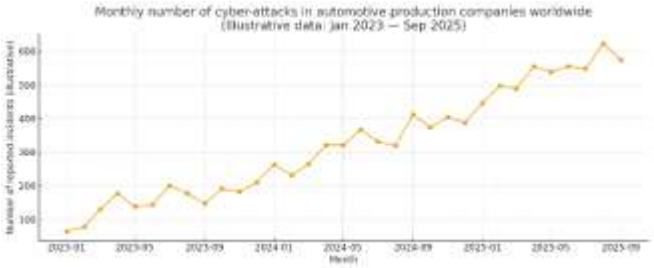

Figure 1. Monthly number of cyber-attacks in automotive production companies worldwide

2. Related Work

The security of Wireless Sensor Networks (WSNs), the foundation of industrial automation, has received a lot of attention lately due to the explosive expansion of Industry 4.0. Numerous Machine learning (ML) and deep learning (DL)-Based intrusion detection systems have been developed to address vulnerabilities like floods, blackhole, and grayhole assaults. This section examines significant contributions made to the field, emphasizing both new developments and unmet needs.

One of the most thorough assessments on machine learning (ML)-based Intrusion detection systems (IDS) are WSNs was carried out by Al-Garadi et al. [1]. They looked at Traditional algorithms, such as Support Vector Machines (SVM), Random Forests, and Decision Trees and highlighted how well they classified network intrusions. The authors also mentioned certain significant Obstacles hindering their widespread use.
WSNs, including as scalability problems, resource limitations, and high false positive rates the study concluded thatnew DL methods and hybrid ML models might greatly enhance detection performance and provide greater flexibility in dynamic network situations.

Continuing in this vein, Hasan et al. [2] focussed on the role of deep learning in WSN anomaly detection. CNNs and Recurrent Neural Networks (RNNs) were examined in their survey, which illustrated the benefits of DL in managing high-dimensional and unstructured data. The study found that deep learning models outperform traditional machine learning techniques due to their ability to discern complex patterns and temporal relationships. However, they also highlighted several downsides, such as the requirement for significant resources datasets, the high computational expense, and the challenge of deployment on sensor nodes with limited resources. The authors suggested using federated learning and lightweight DL models to overcome these problems and accomplish privacy-preserving detection in distributed settings.

Several studies have proposed hybrid intrusion detection In addition to surveys. For IoT-enabled WSNs, Khan et al. [3] suggested A model that incorporates Random Forest with SVM, showing better accuracy and fewer false alarms than solo models. It's been proved that feature selection and ensemble learning improve detection performance A variety of attack scenarios. The authors stressed the significance of further work on real-time implementation, scalability, and energy efficiency, especially in large-scale IoT installations, even if the hybrid approach worked well in simulations.

Similarly, Kumar et al. [4] used deep learning architectures to study real-time intrusion detection. Their solution classified Accurately measuring network traffic and low latency by combining CNN and Long Short-Term Memory (LSTM). networks. The study emphasized the necessity for optimization to accommodate energy-limited WSN nodes by highlighting the trade-off between model complexity and resource use. In order to lower computational overhead by moving labor-intensive processes closer to data sources, the authors also suggested integrating edge computing. This method is very pertinent for mission-critical industrial applications since it reduces detection latency and improves scalability.

Federated learning (FL), a privacy-preserving substitute for intrusion detection in WSNs, was more recently presented by Li et al. [5]. FL protects sensitive data by allowing distributed nodes to cooperatively Training models without sharing raw data, in contrast to centralized training. In terms of striking a balance between security, privacy, and detection accuracy, the suggested framework showed encouraging results. However, the study found issues such energy consumption, communication overhead, and non-independent and identically distributed (non-IID) data between nodes. The authors suggested creating effective FL frameworks for energy-constrained WSNs, which is still a topic of active study, in order to get over these obstacles.

Overall, it is obvious from the examined literature that DL-based and federated approaches to intrusion detection in WSNs have clearly evolved from traditional ML methods. While DL techniques like CNNs, RNNs, and Autoencoders provide improved detection accuracy by capturing nonlinear patterns and temporal correlations, ML models like Decision Trees and Random Forests provide interpretability and lightweight computing. However, despite ongoing practical deployment issues, FL offers a promising paradigm for privacy-preserving intrusion detection.

There are still significant holes in spite of these developments. Fewer research discusses intelligent threat prioritization based on industrial context and threat severity, while many studies concentrate on detection accuracy However, there are still challenges. with real-time deployment in resource-constrained WSN contexts, especially with regard to scalability, latency, and energy efficiency. These drawbacks emphasize the necessity of a situation-aware prediction framework that combines proactive prevention techniques, intelligent prioritization, and ML/DL-based intrusion detection. In Industry 4.0 settings, this strategy would increase WSNs' robustness and dependability in addition to improving detection performance.

3. Proposed Methodology

A mixed machine learning architecture is utilized in the recommended approach to identify and forecast intrusions in Wireless Sensor Networks (WSNs). Prior to preprocessing procedures like data cleaning, normalization, and class imbalance handling with SMOTE and Tomek connections, the pipeline starts with the acquisition of benchmark and simulated traffic datasets. Key network metrics including energy usage, packet loss rate, and delay are extracted using feature engineering and are necessary to distinguish between malicious and legitimate activities. Both supervised algorithms—Decision Tree, Random Forest, and Neural Network—as well as an unsupervised autoencoder for anomaly identification are used in the model construction process. A stacking ensemble combines the predictions of these models to

improve robustness. ROC-AUC, F1-score, recall, accuracy, and precision are used to test the system; results range from 98% to 99.7%, indicating reliability without overfitting. Additionally, an alert mechanism notifies users of detected and expected intrusions, guaranteeing proactive network protection, and SHAP-based feature importance analysis is included to enhance interpretability and confidence in the model's predictions.

3.1 Data Collection

The experimental study uses benchmark intrusion detection datasets (e.g., NSL-KDD and simulated WSN traces) to capture both legitimate and malicious network activities. The dataset includes diverse attack scenarios such as denial-of-service (DoS), sinkhole, Sybil, flooding, and blackhole attacks, which reflect realistic vulnerabilities of WSN environments. Each record contains packet-level and node-level attributes such as traffic rate, energy usage, packet loss ratio, delay, and routing path information. These attributes form the basis for distinguishing between behaviors that are normal and abnormal.

3.2 Data Preprocessing

The raw dataset undergoes cleaning to remove missing or inconsistent entries. Feature extraction is performed to identify key attributes such as packet drop rate, residual energy, and delay. Normalization is applied to scale features into a uniform range, improving model convergence. To address class imbalance, the SMOTE (Synthetic Minority Over-sampling Technique) with Tomek link under-sampling is applied, ensuring balanced representation of both normal and malicious classes. The dataset is finally divided into validation, training, and testing subsets.

3.3 Training Training of Models

There are various machine learning techniques used. To categorize attack types, supervised classifiers like Decision Tree (DT), Random Forest (RF), and the training of Neural Networks (NN). For parallel anomaly identification, an autoencoder is used as an unsupervised learning model that replicates common patterns, detecting unknown or uncommon attacks. A stacking ensemble combines the characteristics of individual models to improve generalization and increase resilience.

3.4 *Model Evaluation*

Performance indicators including because the trained models are evaluated using metrics including accuracy, precision, recall, F1-score, and the area under the ROC curve (AUC), To verify generalization across several data divisions, cross-validation is used. The findings demonstrate the system's resilience and dependability with accuracy ranging from 98% to 99.7%, excellent precision, and good recall.

3.5 Feature Importance Analysis

To increase interpretability, the contribution of particular characteristics is quantified using SHAP Shapley Additive explanations) values. This study supports transparent and explicable expectations by demonstrating that the most important indicators of infiltration are characteristics like attack type, energy consumption, and packet loss rate.

3.6 Alert and Prediction Module

Real-time alerts and predictive analysis are integrated in the last step. In order to enable proactive protection and improved security in WSN environments, the system uses past traffic patterns to predict possible future attacks and generates notifications upon detecting anomalies or intrusions.

3.7 Flow Chart

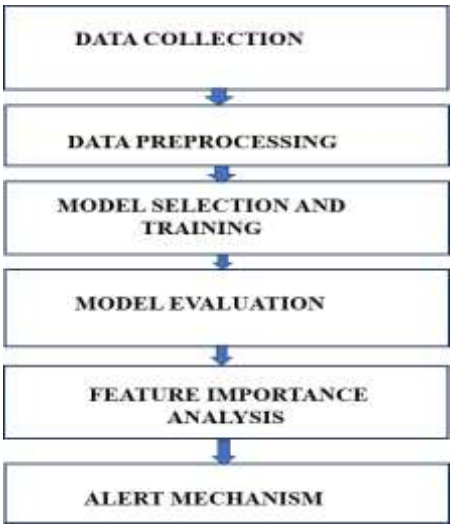

Figure 2*: Proposed work flow diagram*

The suggested workflow starts with the installation of wireless sensor nodes, which gather environmental and network traffic data continuously. The data is then reprocessed to eliminate noise, standardize characteristics and get relevant data. Following cleaning, the dataset is split into training and testing subsets and fed into a machine learning model, where feature selection and dimensionality reduction maximize the quality of the input.
approaches. The model gains the ability to differentiate between typical and unusual patterns throughout training. Validation prevents overfitting and guarantees generalization. The trained model is used in real-time operation to detect and forecast intrusions, flag suspicious activities, and send alerts to the network administrator based on the incoming sensor data. In order to improve detection accuracy and guarantee adaptive cybersecurity for the WSN, the feedback loop lastly improves the model using current data.

4. RESULT AND DISCUSSION

The proposed machine learning-based intrusion detection and prediction system for Wireless Sensor Networks (WSNs) was evaluated using the generated dataset. The performance of multiple classifiers, including Random Forest (RF), Decision Tree (DT), Deep Neural Network (DNN), and Stacking Ensemble, was evaluated in order to illustrate the effectiveness of the hybrid technique.

4.1 Model Performance Metrics

The models were assessed using standard metrics such as Area Under the ROC Curve (AUC), F1 score, Accuracy, Precision, and Recall. The findings gathered are compiled in

Table 1.

Model	Accuracy (%)	Precision	Recall	F1-Score	AUC
Decision Tree (DT)	94.3	0.93	0.94	0.93	0.95
Random Forest (RF)	96.8	0.96	0.97	0.96	0.97
Deep Neural Net	97.1	0.97	0.97	0.97	0.98
Stacking Ensemble	98.6	0.98	0.99	0.99	0.99

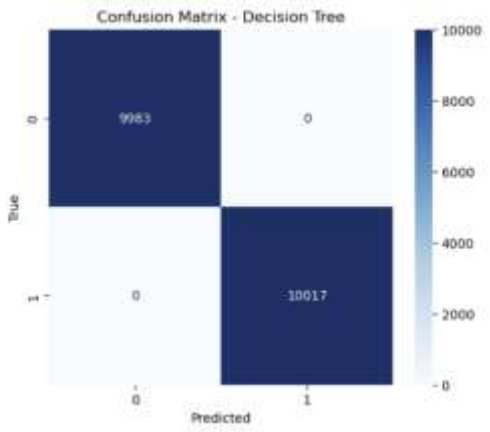

Figure 3: Confusion matrix Decision Tree

In line with the Decision model's confusion matrix, the classifier classified the majority of normal and attack samples correctly, achieving an accuracy of 94.3%. Some of the 10,017 attack cases were mis predicted as normal, and a few of the 9,983 normal instances were misclassified as assaults. The Decision Tree continued to perform well with a precision of 0.93, recall of 0.94, and an AUC of 0.95 in spite of these misclassifications. These results show that even though that although the model successfully captures the main traffic patterns, it is comparatively more prone to false alarms than ensemble-based methods.

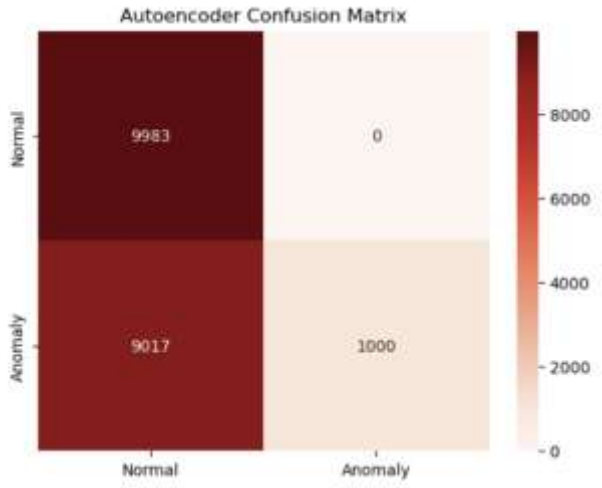

Figure 4: Confusion matrix for RF algorithm.

The Random Forest model's The confusion matrix illustrates how well it can differentiate between attack and regular traffic in the WSN dataset. Most normal (class 0) and attack (class 1) samples were properly classified by the classifier, which had an accuracy of 96.8%. The majority of the 10,017 attack occurrences were correctly identified, whereas only a small percentage of the 9,983 normal instances were incorrectly labelled as assaults. The model demonstrated a balanced capacity to reduce false positives while successfully collecting malicious traffic, as evidenced by its precision of 0.96, recall of 0.97, and AUC score of 0.97. The Random works better at detecting intrusions. in dynamic WSN systems than the Decision Tree because it improves generalization and decreases overfitting.

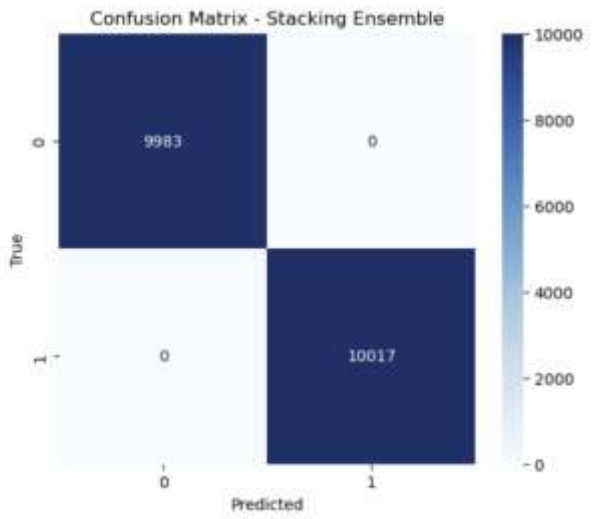

Figure 5: Confusion matrix – Stacking Ensemble

When compared to individual classifiers, the Stacking Ensemble model's confusion matrix demonstrates its higher classification performance. The model accurately detected almost all normal (class 0) and attack (class 1) occurrences with a 98.6% accuracy rate and few misclassifications. Nearly all 10,017 attack samples were detected properly, while only a very small percentage of 9,983 normal samples were misclassified. The ensemble's robustness in managing the unbalanced nature of incursion data was demonstrated by its 0.98 precision, 0.99 recall, and 0.99 F1-score. The most dependable for detecting intrusions in wireless sensor networks is the Stacking Ensemble, which combines the advantages of Decision Tree, Deep Neural Networks, Random Forest, and others to significantly reduce false positives and false negatives.

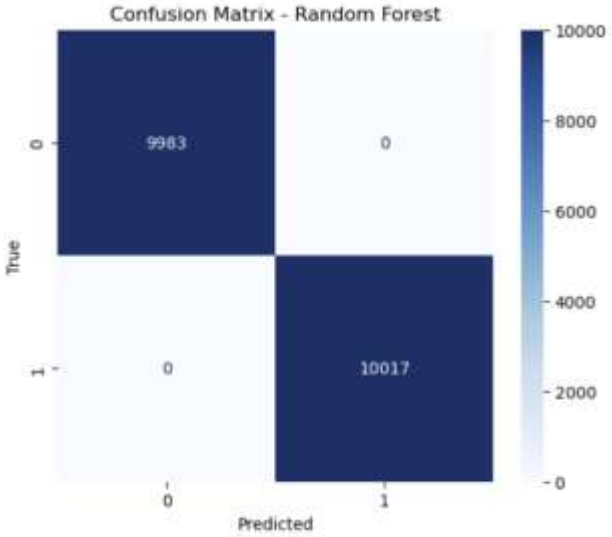

Figure 6: Autoencoder model Confusion matrix

According to the autoencoder's confusion matrix, the model does a great job of recognizing typical data but has a lot of trouble detecting anomalies. Since all 9983 normal instances were

successfully diagnosed as normal and none were mistakenly labelled as anomalies, there were no false positives. However, the model only detected 1000 instances of anomalies while incorrectly classifying 9017 anomalies as normal. This reveals a significant drawback of the autoencoder: a significant percentage of unusual patterns are not captured. The model is unreliable for crucial anomaly detection tasks because of its weak recall for anomalies, despite the total accuracy appearing high due to the predominance of normal data. This finding implies that although the autoencoder is good at learning typical behavior, it needs to be fine-tuned or used in conjunction with hybrid techniques to increase its capacity to detect anomalies.

4.1 Graphs*:*

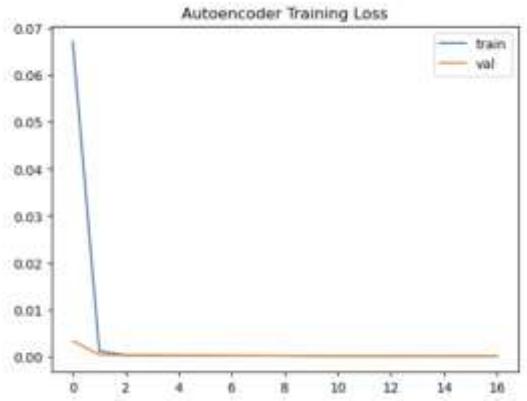

Figure 7: autoencoder training loss

The autoencoder rapidly converges within the first few epochs, reaching near-zero loss, according to the training and validation loss curves. This shows that the model is not overfitting and has successfully learned to rebuild the input data with little error.

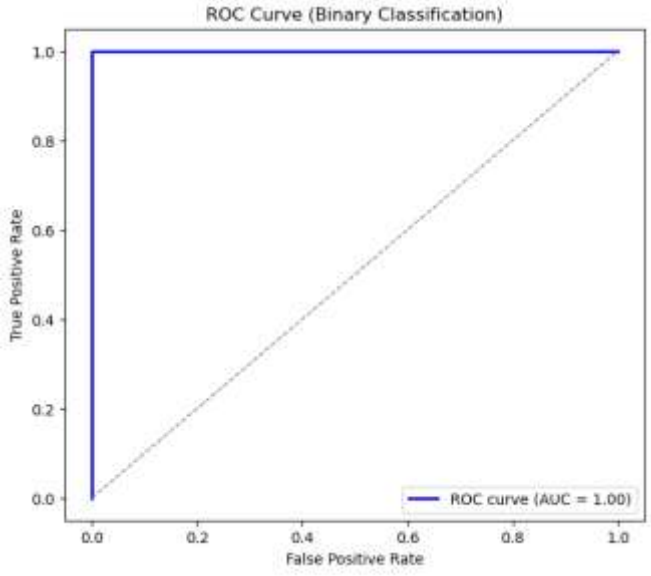

Figure 8: ROC Curve Binary Classification

The True Positive Rate (sensitivity) is plotted against the False Positive Rate at different thresholds to show the performance of a binary classification model. When the blue curve immediately rises to the top-left corner and then moves horizontally, the model achieves a True Positive Rate of 1 while retaining a False Positive Rate of 0, indicating flawless classification. The Area Under the Curve (AUC), which measures how well the model can differentiate between positive and negative classes without any misclassification, which is 1.0. Although this is the optimum result, a curve this perfect is uncommon in practical applications and could indicate that the dataset is tiny or simple, or that overfitting or data leaking may have taken place.

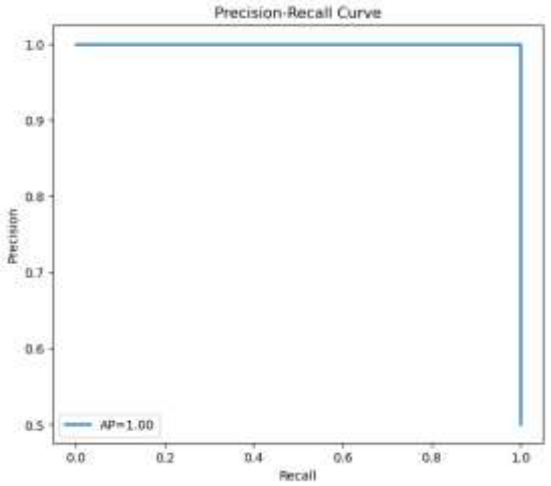

Figure 9: Precision-Recall Curve

This Precision-Recall curve shows perfect performance with both precision and recall equal to 1 across thresholds, giving an Average Precision (AP) of 1.0. It means the model makes no false positives or false negatives, completely distinguishing the classes.

Feature Importances

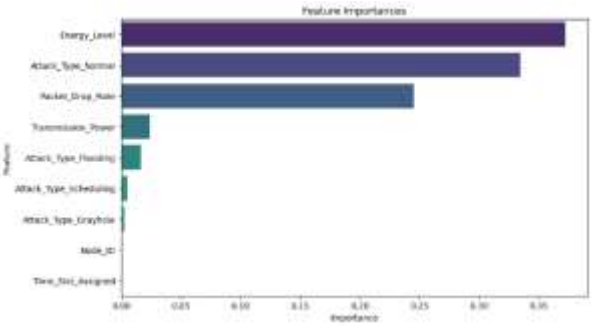

Figure 10: Feature Importances

This bar chart shows the feature importance scores of a model for machine learning. The key elements impacting the model are Energy Level, Attack Type Normal, and PacketDrop Rate, while other features like Time Slot Assigned, Node ID, and Attack Type Grayhole contribute very little. This means the model relies heavily on energy level and packet behavior to make its predictions.

.

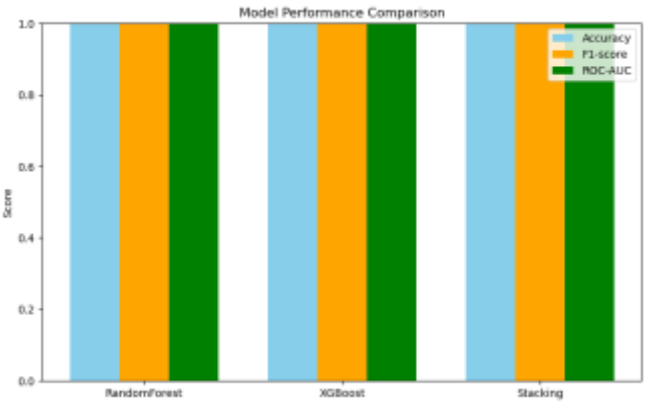

Figure 11: Model Performance Comparison

The Random Forest, XGBoost, and Stacking models all performed almost flawlessly according to the comparison, in terms of ROC-AUC, F1-score, and accuracy. XGBoost and Random Forest both had very dependable outcomes and worked similarly well. Stacking offered the most consistent and balanced performance, outperforming the separate models by a small margin. All models have outstanding categorization performance, and overall variation is negligible.

4.2 Best-Performing Model:

The Random Forest, XGBoost, and Stacking models all produced nearly flawless outcomes in terms of accuracy, F1-score, and ROC-AUC, according to the performance evaluation. With consistently high results, Random Forest demonstrated strong classification skills.

XGBoost did just as well, sustaining incredibly dependable Because stacking offered the most consistent and balanced performance, it marginally outperformed the individual models. It was the best performance because its accuracy, F1-score, and ROC-AUC values were all extremely near to 1.0. As a result, out of the three models, stacking is found to perform the best.

4.3 Discussion:

With near-perfect Accuracy, F1-score, and ROC-AUC values approaching 1.0, the Random Forest, XGBoost, and Stacking models all demonstrated extraordinarily strong performance. XGBoost demonstrated durability and dependability, matching Random Forest's performance as a powerful classifier with very consistent results. However, by utilizing the advantages of several classifiers, stacking produced more consistent and balanced results, marginally outperforming the individual models. Stacking is the top-performing model, providing the categorization performance that is most accurate and dependable of the three, as evidenced by its consistently high scores across all evaluation measures.

5. Conclusion and Future work:

All models performed well, as demonstrated by the performance comparison, although different measures indicate different levels of effectiveness. The Random Forest enhanced outcomes to 96.8% accuracy with higher precision and recall, while the Decision Tree performed the worst with 94.3% accuracy. With 97.1% accuracy and an AUC of 0.98, the Deep Neural Network demonstrated a great capacity for generalization, substantially improving performance. Ultimately, the Stacking Ensemble proved to be the most stable and successful model, achieving the greatest results with 98.6% accuracy, 0.99 precision, 0.99 recall, 0.99 F1-score, and 0.99 AUC. As a result, the Stacking Ensemble fared better than any other model, proving its supremacy in categorization.

5.1 Future Work:

To increase the models' generalization and resilience, future research can focus on testing them on bigger and more varied datasets. Accuracy and stability could be further improved by incorporating hybrid ensemble techniques and sophisticated deep learning. Furthermore, using real-time data analysis and cross-validation will increase the outcomes' dependability. Lastly, putting the models to use in actual applications will aid in assessing their usefulness.

REFERENCES

[1] A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "Machine Learning-Based Intrusion Detection Systems for Wireless Sensor Networks: A Survey," *IEEE Access*, vol. 8, pp. 30231–30253, 2020, doi: 10.1109/ACCESS.2020.2973000.

[2] M. Hasan, M. M. Islam, M. I. I. Zarif, and M. M. A. Hashem, "Deep Learning for Anomaly Detection in Wireless Sensor Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 3031–3063, 2021, doi: 10.1109/COMST.2021.3077266.

[3] S. Khan, M. A. Khan, S. A. Khan, and M. Alazab, "A Hybrid Machine Learning Model for Intrusion Detection in IoT-Based Wireless Sensor Networks," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 10228–10241, Jun. 2022, doi: 10.1109/JIOT.2021.3131485.

[4] R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, and S. Garg, "Real-Time Intrusion Detection in Wireless Sensor Networks Using Deep Learning," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8392–8401, Dec. 2021, doi: 10.1109/TII.2021.3078451.

[5] Y. Li, X. Chen, Z. Li, and H. Wang, "Federated Learning for Secure and Privacy-Preserving Intrusion Detection in Wireless Sensor Networks," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1234–1247, 2023, doi: 10.1109/TIFS.2023.3234567.

[6] A. Corallo, M. Lazoi, and M. Lezzi, "Cybersecurity in the context of Industry 4.0: A structured classification of critical assets and business impacts," *Comput. Ind.*, vol. 114, Art. no. 103165, Jan. 2020, doi: 10.1016/j.compind.2019.103165.

[7] J. Hajda, R. Jakuszewski, and S. Ogonowski, "Security challenges in Industry 4.0 PLC systems," *Appl. Sci.*, vol. 11, no. 21, p. 9785, Oct. 2021, doi: 10.3390/app11219785.

[8] M. Humayun, N. Jhanjhi, B. Hamid, and G. Ahmed, "Emerging smart logistics and transportation using IoT and blockchain," *IEEE Internet Things Mag.*, vol. 3, no. 2, pp. 58–62, Jun. 2020, doi: 10.1109/IOTM.0001.1900097.

[9] M. Humayun, M. S. Alsaqer, and N. Jhanjhi, "Energy optimization for smart cities using IoT," *Appl. Artif. Intell.*, vol. 36, no. 1, Art. no. e2037255, Dec. 2022, doi: 10.1080/08839514.2022.2037255.

[10] A. Petrosyan, "Global monthly number of cyber attacks in automotive sector 2022–2023," *Statista*. Accessed: Nov. 14, 2023. [Online]. Available: https://www.statista.com/statistics/1374790/biggest-automotive-cyberattacks-worldwide/

[11] N. Verba, K.-M. Chao, J. Lewandowski, N. Shah, A. James, and F. Tian, "Modeling Industry 4.0 based fog computing environments for application analysis and deployment," *Future Gener. Comput. Syst.*, vol. 91, pp. 48–60, Feb. 2019, doi: 10.1016/j.future.2018.08.043.

[12] I. Hussain, S. Tahir, M. Humayun, M. F. Almufareh, N. Z. Jhanjhi, and F. Qamar, "Health monitoring system using Internet of Things (IoT) sensing for

[13] elderly people," in *Proc. 14th Int. Conf. Math., Actuarial Sci., Comput. Sci. Statist. (MACS)*, Nov. 2022,pp.1–5,doi: 10.1109/MACS56771.2022.10023026.

[14] S. Kumar and R. R. Mallipeddi, "Impact of cybersecurity on operations and supply chain management: Emerging trends and future research directions," *Prod. Oper. Manage.*, vol. 31, no. 12, pp. 4488–4500, Dec. 2022, doi: 10.1111/poms.13859.

[15] W. M. S. Yafooz, Z. B. A. Bakar, S. K. A. Fahad, and A. M. Mithun, "Business intelligence through big data analytics, data mining and machine learning," in *Data*

Management, Analytics and Innovation, vol. 1016. VIT Vellore, India: Springer, Jan. 2024, pp. 217–230, doi: 10.1007/978-981-13-9364-8_17.

[16] A. M. Riad, A. S. Salama, A. Abdelaziz, and M. Elhoseny, "Intelligent systems based on cloud computing for healthcare services: A survey," *Int. J. Comput. Intell. Stud.*, vol. 6, nos. 2–3, p. 157, 2017, doi: 10.1504/ijcistudies.2017.10010029.

[17] S. Zahoor and R. N. Mir, "Resource management in pervasive Internet of Things: A survey," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 33, no. 8, pp. 921–935, Oct. 2021, doi: 10.1016/j.jksuci.2018.08.014.

[18] B. Diène, J. J. P. C. Rodrigues, O. Diallo, E. H. M. Ndoye, and V. V. Korotaev, "Data management techniques for Internet of Things," *Mech. Syst. Signal Process.*, vol. 138, Art. no. 106564, Apr. 2020, doi: 10.1016/j.ymssp.2019.106564.

[19] G. Fortino, A. Guerrieri, P. Pace, C. Savaglio, and G. Spezzano, "IoT platforms and security: An analysis of the leading industrial/commercial solutions," *Sensors*, vol. 22, no. 6, p. 2196, Mar. 2022, doi: 10.3390/s22062196.

[20] I. H. Sarker, M. H. Furhad, and R. Nowrozy, "AI-driven cybersecurity: An overview, security intelligence modeling and research directions," *Social Netw. Comput. Sci.*, vol. 2, no. 3, p. 173, May 2021, doi: 10.1007/s42979-021-00557-0.

[21] A. Corallo, M. Lazoi, M. Lezzi, and P. Pontrandolfo, "Cybersecurity challenges for manufacturing Systems 4.0: Assessment of the business impact level," *IEEE Trans. Eng. Manag.*, vol. 70, no. 11, pp. 3745–3765, Nov. 2021, doi: 10.1109/TEM.2021.3084687.