Ensemble Learning with Sleep Mode Management to Enhance Anomaly Detection in IoT Environment

Khawlah Harahsheh[1], Rami Al-Naimat[2], Malek Alzaqebah[3], Salam Shreem[4], Esraa Aldreabi[5], Chung-Hao Chen[6]
Ph.D. Student, Department of Electrical and Computer Engineering, Old Dominion University, Norfolk, VA, 23529 USA[1]
Independent Scholar, Karak, Jordan[2]
Department of Mathematics-College of Science, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia[3]
Basic and Applied Scientific Research Center, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia[3]
Independent Scholar, Chicago, USA[4]
Independent Scholar, New York, USA[5]
Department of Electrical and Computer Engineering, Old Dominion University, Norfolk, VA, 23529 USA[6]

***Abstract*—The rapid proliferation of Internet of Things (IoT) devices has underscored the critical need for energy-efficient cybersecurity measures. This presents the dual challenge of maintaining robust security while minimizing power consumption. Thus, this paper proposes enhancing the machine learning performance through Ensemble Techniques with Sleep Mode Management (ELSM) approach for IoT Intrusion Detection Systems (IDS). The main challenge lies in the high-power consumption attributed to continuous monitoring in traditional IDS setups. ELSM addresses this challenge by introducing a sophisticated sleep-awake mechanism, activating the IDS system only during anomaly detection events, effectively minimizing energy expenditure during periods of normal network operation. By strategically managing the sleep modes of IoT devices, ELSM significantly conserves energy without compromising security vigilance. Moreover, achieving high detection accuracy with limited computational resources poses another problem in IoT security. To overcome this challenge, ELSM employs ensemble learning techniques with a novel voting mechanism. This mechanism integrates the outputs of six different anomaly detection algorithms, using their collective intelligence to enhance prediction accuracy and overall system performance. By combining the strengths of multiple algorithms, ELSM adapts dynamically to evolving threat landscapes and diverse IoT environments. The efficacy of the proposed ELSM model is rigorously evaluated using the IoT Botnets Attack Detection Dataset, a benchmark dataset representing real-world IoT security scenarios, where it achieves an impressive 99.97% accuracy in detecting intrusions while efficiently managing power consumption.**

Keywords*—*IoT; IDS; machine learning; ensemble technique; sleep-awake cycle; cybersecurity; anomaly detection

I. Introduction

The Internet of Things (IoT) encompasses many devices, from small sensors to larger, more complex machines. These devices often have specific characteristics and limitations concerning power capacity and time consumption, which are crucial to consider in their design and deployment. Examples of these limitations include power capacity limitations, time-consuming processes, and capacity constraints [1]. Integrating IoT devices into daily life has raised concerns about power consumption and security. Indeed, many IoT devices run outdated firmware, rely on old legacy protocols, and have constrained computational resources. These devices are prone to failure and are vulnerable to a wide spectrum of anomalies, such as malicious attacks, traffic congestion, connectivity problems, and flash crowds [2].

Nevertheless, the high resource requirements of complex and heavy-weight conventional security mechanisms cannot be afforded by (a) the resource-constrained IoMT edge devices with limited processing power, storage capacity, and battery life, and/or (b) the constrained environment in which The IoMT devices are deployed and interconnected using lightweight communication protocols [3]. Traditional security measures often require continuous device operation, leading to excessive power usage. A solution that allows IoT devices to remain in sleep mode until necessary can significantly reduce energy consumption while maintaining security.

IoT has revolutionized the way people interact with technology, connecting a myriad of devices to the Internet. However, this interconnectivity poses significant security challenges. IoT devices are often resource-constrained and become easy targets for cyber-attacks, making robust security mechanisms crucial for maintaining system integrity and user privacy. The wide range of different communication technologies (e.g., WLANs, Bluetooth, Zigbee) and types of IoMT devices (e.g., medical sensors, actuators) incorporated in IoMT edge networks are vulnerable to various types of security threats. This raises many security and privacy challenges for such networks, as well as for the healthcare systems relying on these networks [4].

Software-defined networking (SDN) is a modern paradigm that enhances network management through its dynamic and programmable architecture [5]. However, SDN lacks inherent security features, and one significant issue that may impede its widespread adoption is the potential for novel assaults [6]. Characteristics such as network programmability and centralized control introduce new faults and vulnerabilities, opening the door to threats that did not previously exist [7, 8]. The literature identifies seven potential attack vectors against SDNs, with three specifics to SDN networks [9]. A key countermeasure to secure SDN is the deployment of Intrusion Detection Systems (IDS), which can identify and mitigate

malicious activities in real-time by monitoring network traffic [10]. IDS plays a pivotal role in identifying and mitigating cyber threats in IoT environments [11]. By monitoring network traffic and analyzing system behavior, IDS can detect potential threats before they cause harm. However, traditional IDS solutions often require significant computational resources, leading to high power consumption and latency, which are impractical for many IoT devices [12].

Traditional IDS solutions are often characterized by their high-power consumption, primarily due to the intensive computational processes involved in monitoring and analyzing network traffic [13]. This poses a significant challenge for IoT environments, where devices are designed to operate with minimal energy use. The deployment of power-intensive IDS can lead to rapid battery depletion, reducing the operational lifespan of IoT devices. This creates a critical trade-off situation: ensuring robust security through comprehensive IDS capabilities versus maintaining the energy efficiency critical to IoT device functionality.

Given the resource constraints of IoT devices, developing Intrusion Detection Systems (IDS) that efficiently manage time and power consumption is critical for practical deployment. This entails ensuring prolonged battery life and quick response times, striking a balance between robust security measures and minimal resource usage. In various domains like risk control, fraud detection, and decision-making, there's a significant focus on identifying unexpected events or patterns from datasets. Both static IoT systems (e.g., smart homes, smart buildings) and dynamic IoT networks (e.g., Vehicular Ad-hoc Networks) incorporate numerous lightweight, resource-constrained devices. Thus, efforts to develop IDS for IoT environments must optimize both intrusion detection effectiveness and resource efficiency to seamlessly integrate and widely adopt IDS within IoT ecosystems [14].

Machine learning (ML) offers promising avenues for enhancing the efficiency of IDS in IoT. By using ML algorithms, IDS can adapt to new threats more efficiently and reduce the time and power required for data processing and threat detection. This paper aims to explore the implementation of machine learning techniques in improving time and power efficiency in IoT intrusion detection systems. Energy efficiency is increased when a device has a sleep-wake mechanism, which enables it to go into sleep mode while not in use. The sleep-wake cycle in IoT devices is a critical feature designed to balance energy consumption with operational functionality. This cycle allows devices to conserve power by entering a low power 'sleep' mode when active operation is not needed and 'waking up' to a fully operational state when required to perform tasks. Additionally, the module integrates an isolation forest with an autoencoder, combining the best features of both techniques to provide enhanced anomaly detection in Internet of Things devices, even in sleep mode.

In this paper, a novel hybrid model designed to conserve power within IoT environments is introduced. This model incorporates a mechanism to keep warning devices inactive until abnormal traffic is detected, thereby reducing unnecessary power consumption. Our proposed module leverages ensemble learning and a strategic sleep-wake protocol to optimize power usage while maintaining robust anomaly detection capabilities.

By integrating multiple machine learning models, such as Logistic Regression (LR), Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and XGBoost, and using an ensemble learning and voting mechanism, the module enhances detection accuracy. The sleep-wake mechanism ensures that IoT devices remain in a low-power state until an anomaly is detected, significantly conserving energy. The main contributions of this work can be summarized as follows:

- Enhanced anomaly detection with hybrid ensemble learning: A novel hybrid model integrating multiple machine learning models with a voting mechanism is introduced to improve the accuracy of anomaly detection in IoT systems.
- Optimized power consumption with sleep-wake mechanism integration: This allows devices to conserve energy by entering sleep states when not actively needed, thereby extending battery life and device operation time.
- Improved computational efficiency with caching: A caching mechanism for the fastest-performing model within the ensemble is introduced to reduce processing time for future detections and improve real-time application performance.

The paper is structured as follows: Section II provides an overview of related work and compares the main differences between our work and other existing research. In Section III, the research methodology details the experimental configuration, and the system flowchart is presented. The experimental results and analysis are thoroughly described in Section IV. Future work is articulated in Section V, followed by a conclusion in Section VI.

II. Related Work

In recent years, significant advancements have been made in developing Intrusion Detection Systems (IDS) for secure IoT environments. The integration of machine learning techniques has greatly enhanced the ability to detect and mitigate cyber threats. Various approaches have been proposed to improve the detection accuracy and efficiency of IDS, especially using ensemble learning methods.

An ensemble learning approach that integrates logistic regression, decision trees, and gradient boosting to enhance the performance of IDS was proposed in study [15]. This method addresses challenges in accuracy, speed, false alarms, and unknown attack detection. Using the CSE-CIC-IDS2018 dataset and Spearman's rank correlation, 23 out of 80 features were selected. The proposed model achieved 98.8% accuracy, demonstrating its effectiveness in improving data security.

Another ensemble-based intrusion detection model was proposed in study [16] using multiple machine learning techniques, such as Decision Trees, J48, and SVM. Particle swarm optimization was used to select the nine most relevant

features in the KDD99 dataset, resulting in a model with 90% accuracy.

Additionally, a hybrid IDS model based on Naive Bayes and SVM was presented in study [17]. The study normalized and preprocessed a real-time historical log dataset, enhancing the model to achieve 95% accuracy and precision. The addition of session-based features further increased classifier performance.

Performance analysis of multiple classical machine learning algorithms on several ID-based datasets, including CICIDS2018, UNSW-NB15, ISCX2012, NSLKDD, and CIDDS001, was conducted in study [18]. Techniques such as SVM, k-nearest Neighbors, and Decision Trees were deployed, with Decision Trees outperforming other classifiers, achieving detection accuracy rates between 99% and 100% for all datasets.

A lightweight IDS developed using SVM aimed to detect unknown and misuse attempts in IoT networks [19]. Experiments on different functions, such as linear, polynomial, and radial basis, showed reduced processing time and complexity due to selected features. However, the algorithm struggled to detect intrusions without affecting traffic flow rates.

A framework for botnet attack detection using a sequential detection architecture was introduced, reducing processing resource demand through relevant feature selection [20]. The N-BaIoT dataset was used, achieving 99% detection performance with Decision Trees (DT), Naive Bayes (NB), and Artificial Neural Networks (ANN). Hybrid classification was used in each sub-engine to enhance accuracy.

Alzaqebah et al. [21] developed a Network Intrusion Detection System (NIDS) using a modified Grey Wolf Optimization algorithm to enhance the efficiency of IDS. This approach smartly grouped wrapper and filter methods to include informative features in every iteration, combined with an Extreme Learning Machine (ELM) for faster classification.

Additionally, Ugendhar et al. [22] proposed an IDS that uses a deep multilayer classification approach. This system incorporates an autoencoder with a reconstruction feature to perform dimensionality reduction. The developed deep multilayer classification approach uses the autoencoder to reduce the dimensionality of the reconstruction feature, enhancing the efficiency and accuracy of the IDS.

In study [23], Grey Wolf Optimization (GWO) was proposed for feature selection, combined with Particle Swarm Optimization (PSO) to optimize the updating process for each grey wolf position. This hybrid approach harnesses PSO's ability to preserve the individual's best position information, which helps prevent GWO from falling into local optima. The performance of this technique is verified using the NSL-KDD dataset. Classification is conducted using k-means and SVM algorithms, and performance is measured in terms of accuracy, detection rate, false alarm rate, number of features, and execution time.

In another study, Samriya and Kumar [24] used a fuzzy-based Artificial Neural Network (ANN) for developing an IDS. They employed the Spider Monkey Optimization algorithm for dimensionality reduction and tested the model with the NSL-KDD dataset. Moreover, an ensemble approach combining multiple machine learning models, such as Decision Tree (DT), Naive Bayes (NB), and Support Vector Machine (SVM), has been shown to enhance the detection capabilities of IDS. This method leverages the strengths of individual classifiers to improve overall system performance.

An Ensemble learning-based method was proposed in study [25], combining Isolation Forest and Pearson's Correlation Coefficient to reduce computational cost and prediction time. The Random Forest classifier was used to enhance performance. Evaluations on Bot-IoT and NF-UNSW-NB15-v2 datasets showed RF-PCCIF and RF-IFPCC achieving up to 99.99% accuracy and prediction times as low as 6.18 seconds, demonstrating superior performance.

A study proposed in [26] developed an IDS for CAN bus networks using ensemble techniques and the Kappa Architecture. The IDS combines multiple machine learning classifiers to enhance real-time attack detection. Supervised models were developed and improved with ensemble methods. Evaluation of common CAN bus attacks showed the stacking ensemble technique achieving an accuracy of 98.5%.

A novel approach to enhance intrusion detection was proposed in study [27]. It begins with denoising data to address the imbalance, followed by employing the enhanced Crow search algorithm for feature selection. An ensemble of four classifiers then classifies intrusions. Evaluation of NSL-KDD and UNSW-NB15 datasets shows accuracy rates of 99.4% and 99.2%, respectively, highlighting superior performance compared to existing methods.

Table I provides a detailed analysis of how researchers have used machine learning algorithms for intrusion detection across different datasets. It explores the specific techniques employed in each study, the datasets leveraged for training and evaluation, and any noteworthy remarks about the methodology or findings.

While many studies employ classic machine learning techniques, there is a notable lack of approaches addressing power efficiency in IoT environments, which is critical given the resource constraints of IoT devices. Moreover, these methods often fail to incorporate mechanisms for real-time intrusion detection and efficient computational performance, which is crucial for practical deployment in dynamic IoT networks.

The proposed Ensemble Learning with Sleep Mode Management (ELSM) aims to fill these gaps by using a more recent IoT botnet dataset, offering a contemporary perspective on current security challenges. The proposed method achieves a high detection rate by employing an ensemble learning technique with a novel voting mechanism while enhancing robustness and efficiency. This is particularly important in IoT environments where power conservation is crucial; Hence, integrating sleep-awake management optimizes power usage. Additionally, our approach accelerates the detection process by caching the fastest module for subsequent iterations, thereby improving computational efficiency.

TABLE I. A COMPREHENSIVE REVIEW OF MACHINE LEARNING TECHNIQUES FOR INTRUSION DETECTION IN VARIOUS DATASETS

Reference	Dataset	Techniques Used	Acc	Remarks
[15]	CSE-CIC-IDS2018	Logistic Regression, Decision Trees, Gradient Boosting	98.8%	Uses Spearman's rank correlation.
[16]	KDD99	Decision Trees, J48, SVM	90%	Employs Particle Swarm Optimization for feature selection.
[17]	Real-time historical log	Naive Bayes, SVM	95%	Data normalized and preprocessed before applying machine learning algorithms.
[18]	CICIDS2018, UNSW-NB15, ISCX2012, NSL-KDD, CIDDS001	SVM, k-Nearest Neighbors, Decision Trees	between 99% and 100%	Evaluates multiple classical machine learning algorithms on various datasets.
[19]	CICIDS2017	SVM (linear, polynomial, radial basis functions)	98.03%	Focuses on detecting denial-of-service attacks using simple features (e.g., packet arrival rates).
[20]	N-BaIoT	Artificial Neural Network, J48 Decision Tree, Naive Bayes	99%	Reduces processing demands by selecting relevant features.
[21]	NSL-KDD	Grey Wolf Optimization, ELM (Extreme Learning Machine)	99.12%	Focuses on achieving fast classification using ELM.
[22]	NSL-KDD	A deep multilayer classification approach	96.7%	Autoencoder Employed for dimensionality reduction.
[23]	NSL-KDD	The classification is done using the k-means and SVM algorithms	-	Hybrid Grey Wolf Optimizer with Particle Swarm Optimization for feature selection.
[24]	NSL-KDD	Fuzzy ANN (Artificial Neural Network), Spider Monkey Optimization	98.70%	Utilizes fuzzy logic for potentially enhanced accuracy.
[25]	Bot-IoT, NF-UNSW-NB15-v2	Isolation Forest, Pearson's Correlation Coefficient, Random Forest	99.99%	Utilizes Isolation Forest for anomaly detection and feature selection with Random Forest for classification.
[26]	CAN bus attack datasets	Random Forest, Decision Tree, XGBoost	98.5%.	Uses ensemble learning techniques with Kappa Architecture for improved performance.
[27]	NSL-KDD, UNSW-NB15	SVM, k-nearest Neighbors, Random Forest, Long Short-Term Memory (LSTM)	99.4% and 99.2%	Applies data denoising and Crow search algorithm for feature selection and explores LSTM for deep learning.
[28]	CIC-IDS 2017	Improved Golden Jackal Optimizer, ANN	98.60%	Applies deep learning and optimization techniques for intrusion detection in smart city environments.
[29]	NSL-KDD	Random Forest, Naive Bayes, J48	99.10%	Investigates the effectiveness of hybrid classification methods.

III. PROPOSED METHOD

Power management is critical in the world of IoT devices, with a special focus on leveraging sleep mode to enhance energy efficiency. This is particularly crucial for devices operating on limited resources such as batteries. Sleep-wake cycles are essential to prolong operational life and increase longevity by reducing active mode duration, thus lowering maintenance demands and costs. In networked IoT environments, timely activation and communication can mitigate network congestion and minimize communication overhead. Key design considerations include optimizing wake-up frequency to align with specific application needs and power constraints, minimizing wake-up duration to the minimum for task completion before returning to sleep mode, and ensuring device responsiveness and reliability to guarantee timely wakeups.

Anomaly detection in IoT frameworks is crucial for the early identification of issues ranging from minor system faults to major cybersecurity vulnerabilities, thereby enhancing system reliability, security, and operational efficiency. Machine learning is essential in this context, leveraging historical data to detect patterns and anomalies. In IoT, anomalies manifest as deviations from expected behaviors, signaling potential system failures, security breaches, or environmental changes. Anomalies can be categorized into point anomalies, which are significant deviations in individual data points, context anomalies specific to conditions, and collective anomalies, where seemingly normal data points together indicate suspicious activity. Each type requires distinct detection methodologies, with machine learning playing a pivotal role in their identification and resolution.

The proposed hybrid model aims to reduce power consumption in IoT environments by optimizing usage through a strategic sleep-wake protocol activated exclusively during anomaly detection phases. This methodology employs a composite model merging an autoencoder with an isolation forest, offering dual benefits of increased energy efficiency and enhanced detection precision. The model's construction, illustrated in Fig. 1, includes steps designed to conserve power and enhance IDS compatibility with IoT device limitations. Moreover, deploying this model at the edge is preferable to using cloud resources, as the primary goal is power preservation, while cloud data transmission consumes additional energy. Cloud computing may not be entirely suitable for provisioning IoT applications [29], primarily due to connectivity challenges between cloud resources in the core network and edge devices [30].

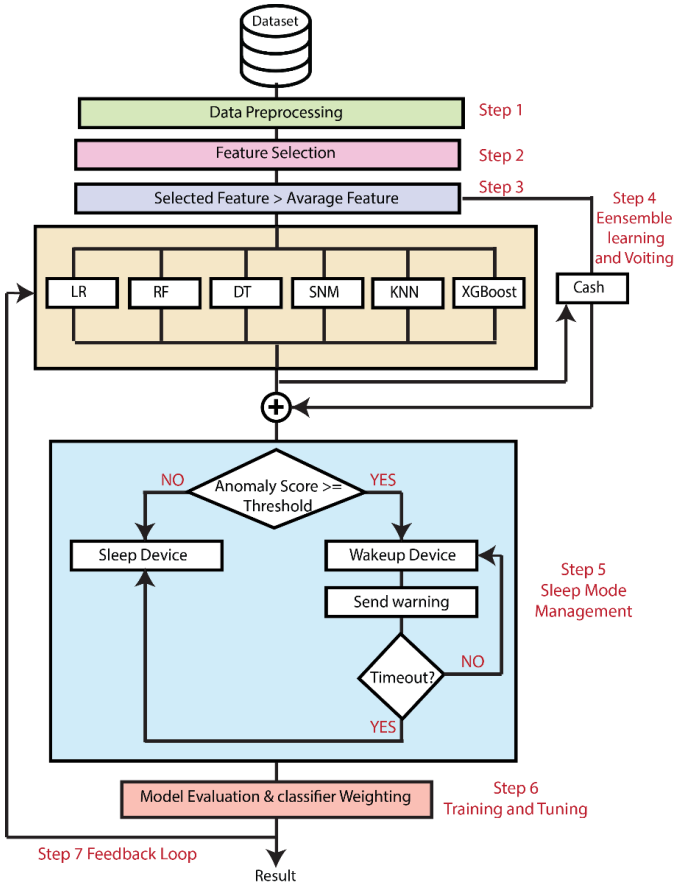

Fig. 1. The flowchart of the ELSM module.

The flowchart in Fig. 1 illustrates a comprehensive methodology for enhancing anomaly detection in IoT environments by leveraging ensemble learning and incorporating a sleep mode management mechanism. The process is divided into several key steps, each contributing to optimizing power efficiency and improving detection accuracy. The steps for Ensemble Learning with Sleep Mode Management (ELSM) are as follows:

1) Data preprocessing: Data collection and preprocessing from IoT sensors and devices are critical steps to ensure the suitability of data for machine learning analysis. Initially, data is gathered from sensors embedded in IoT devices, capturing real-time information about the environment or the device itself. This raw data then undergoes preprocessing, which involves cleaning, filtering, and transforming it into a format suitable for analysis. Tasks in preprocessing may include removing noise, missing handling values, normalizing data, and extracting features. Additionally, data may be aggregated or sampled to reduce dimensionality and enhance computational efficiency. By meticulously preparing the data and ensuring its quality and relevance, subsequent machine learning algorithms can effectively derive meaningful insights and support informed decision-making in IoT applications.

2) Feature selection: In the proposed methodology, a filter method is employed to enhance the performance of the Intrusion Detection System (IDS). Improving speed is crucial for minimizing power consumption, reducing the reliance on extensive computational resources, and achieving faster processing times. Filter methods are noted for their swift execution compared to other feature selection techniques [32].

3) Feature voting: Selected features that surpass a specified average threshold are advanced to the next phase. This ensures that only the most significant features are used, thereby maintaining system efficiency and ensuring optimal feature selection for further analysis and performance optimization. The Feature Voting process involves ranking features based on their contribution to the classification task. Features with important scores above a predefined average threshold are considered significant and selected for further analysis. This reduction is crucial for maintaining high accuracy while ensuring computational efficiency, particularly in resource-constrained IoT environments.

4) Ensemble learning and voting: This step involves training multiple machine-learning models, including Logistic Regression (LR), Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and XGBoost. These models are combined using ensemble learning techniques to enhance overall prediction accuracy. The ensemble method typically employs a voting mechanism, where each model votes on the presence of an anomaly, and the final decision is based on the majority vote. Each model was evaluated based on two criteria: accuracy and prediction time. For each model, the training time and prediction time per instance are calculated as in Eq. (1), Eq. (2), and Eq. (3) respectively.

$$t_{\text{train},i} = \text{time\_end}\, d_{\text{train},i} - \text{time\_start}_{\text{train},i} \quad (1)$$

$$t_{\text{pred},i} = \frac{\text{time\_end}_{\text{pred},i} - \text{time\_start}_{\text{pred},i}}{n_{\text{test}}} \quad (2)$$

$$Accuracy_i = \frac{\sum_{j=1}^{n_{\text{tot}}} \mathbf{1}(\hat{y}_j = y_j)}{n_{\text{test}}} \quad (3)$$

where, 1 is the indicator function that returns 1 if the predicted label $\hat{y}_j$ matches the true label yj, and $n$test is the number of test instances. The weight and the normalized weight for each module were then calculated separately, as shown in Eq. (4) and Eq. (5), respectively.

$$w_i = \frac{Accuracy}{t_{pred,i}} \quad (4)$$

$$w_i' = \frac{w_i}{\sum_{k=1}^{6} w_k} \quad (5)$$

Afterwards, the ensemble model which combines the predictions of all individual models using a weighted voting mechanism was calculated. For a given instance x, the ensemble prediction $\hat{y}_{\text{ensemble}}$ as in Eq. (6):

$$\hat{y}_{\text{ensemble}} = arg\max_{c} \sum_{i=1}^{6} w_i' \cdot P_i(y = c \mid x) \quad (6)$$

Where $(y=c|x)$ is the probability predicted by model mi that the instance x belongs to class c. Finally, the accuracy is calculated for the ensemble techniques as in Eq. (7).

$$Accuracy_{ensemble} = \frac{\sum_{j=1}^{n_{tet}} \mathbf{1}\left(\hat{y}_{ensemble,j} = y_j\right)}{n_{test}} \quad (7)$$

5) Sleep mode management: The sleep-wake cycle plays a crucial role in managing energy consumption in devices, particularly in battery-powered or energy-constrained environments. During sleep mode, the device conserves energy by shutting down non-essential functions, maintaining only vital components in a significantly reduced power state. Triggers such as scheduled timers, external signals such as motion detection, or internal alerts such as critical battery levels prompt the device to transition from sleep to wake mode. In wake mode, the device resumes full functionality, enabling tasks such as sensing, data processing, and communication. Once these tasks are completed, the device can return to sleep mode, ensuring efficient power preservation. This cycle optimizes energy usage, thus enhancing device efficiency and longevity.

6) Training and tuning: Testing and validation of the system under real-world conditions are essential to confirm its effectiveness in anomaly detection and power efficiency. During testing, the system's ability to accurately identify anomalies across various scenarios and environmental conditions is evaluated, providing insights into its robustness and reliability. Additionally, efforts are focused on assessing the system's power efficiency to ensure optimal operation within IoT device constraints, while minimizing energy consumption without compromising detection accuracy. By rigorously testing performance metrics and conducting thorough validation procedures, the system's suitability for deployment in practical applications is affirmed, instilling confidence in its ability to enhance security and efficiency in IoT environments.

7) The feedback loop: The feedback loop for continuous improvement in the system is pivotal for enhancing its performance over time. This iterative process ensures that the system remains agile and responsive to changes in its environment, thereby improving accuracy in anomaly detection. Additionally, the feedback mechanism facilitates self-optimization based on performance metrics and energy consumption data, allowing for adjustments to algorithms and configurations as needed. This iterative feedback loop enhances the system's overall effectiveness while promoting efficient energy utilization, ensuring sustainable operation in IoT environments.

IV. Experimental Results and Discussion

This section presents the experimental results and analysis of the proposed model, which aims to enhance anomaly detection in IoT environments through ensemble learning and a strategic sleep-wake mechanism. Additionally, the process crucial for maintaining high accuracy while ensuring efficiency in the resource-constrained nature of IoT environments is demonstrated.

A. Dataset

The dataset utilized in this research is known as the InSDN dataset, published in 2020 by M. Elsayd et al. [31]. The primary objective of creating the InSDN dataset was to reduce its size compared to other IDS datasets while providing a realistic representation of traffic in an SDN environment. Unlike NSL-KDD and KDD99, InSDN includes real-world network traffic collected from an SDN environment and categorizes it based on the presence of DDoS attacks [32].

The dataset comprises a total of 343,889 records with 84 features. Of these, 127,828 records correspond to normal traffic, while 216,061 records represent attack traffic in both OSV and metasploitable files. InSDN encompasses various attack scenarios, including SYN, TCP, UDP, and ICMP floods, as well as Slowloris attacks. The distribution of samples within the InSDN dataset is detailed in Fig. 2.

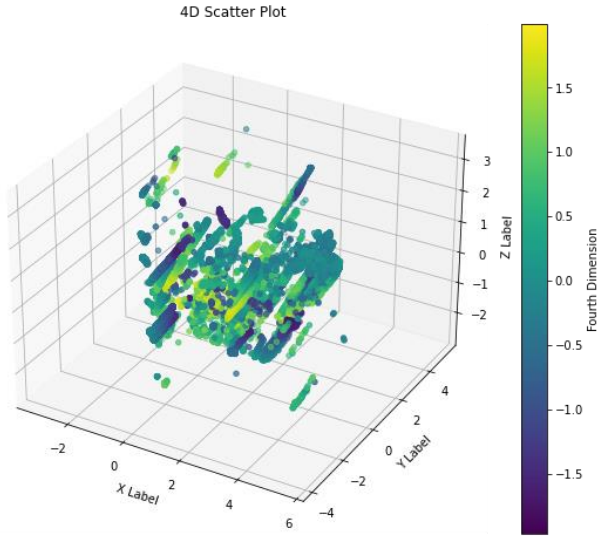

Fig. 2. 4D scatter plot for records on InSDN dataset.

B. Experimental Result and Analysis

The study employed a Random Forest classifier trained on the dataset to evaluate feature importance and classification accuracy. The dataset was split into training and testing sets, achieving an impressive accuracy of 99.97%. Through analysis, the top features contributing to classification were identified, ranked by importance in descending order. Fig. 3 presents a bar chart visualizing the importance of each feature, offering valuable insights into which features are most critical for the model's decisions.

Features with an average importance score above 0.0556 were selected for further analysis, as depicted in Fig. 4, effectively reducing the dataset's dimensionality. This focused approach enabled us to focus on the most relevant features, optimizing the model's performance and computational efficiency. Such selection is crucial for maintaining high accuracy while ensuring the system remains efficient and suitable for the resource-constrained nature of IoT environments.

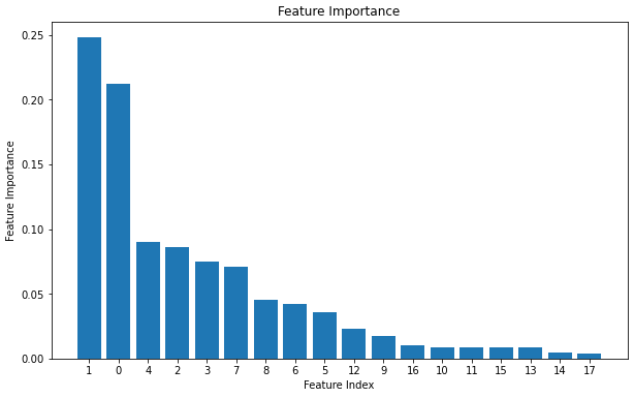

Fig. 3. The importance of the selected features.

After feature selection, an ensemble model was developed by combining Logistic Regression, Decision Tree, and SVM to leverage the strengths of multiple classifiers, as explained in Section III, Step 4.

As a result, Logistic Regression achieved an accuracy of 99.99% with a prediction time of 1.88e-7 seconds, weighted at 871.5946. The Decision Tree model exhibited 99.96% accuracy with a prediction time of 11.16e-7 seconds, having the highest weight of 5789.0159. SVM achieved an accuracy of 99.98% with a prediction time of 0.00011 seconds, weighted at 167.8950. Details of other classifier results are provided in Table II.

As visualized in Fig. 5, ELSM emerges as a robust performer across multiple key metrics. It achieved an accuracy of 99.97%, closely aligning with top-performing models such as LR, SVM, and XGBoost, which scored 99.99%. ELSM's precision of 94.97% underscores its ability to accurately identify positive instances. Moreover, ELSM achieved a recall of 93.93%, demonstrating its effectiveness in capturing the most actual positive instances. Its F1 score of 94.31% strikes a balance between precision and recall, ensuring robust overall performance.

C. Discussion

ELSM excels with a perfect ROC AUC of 100.00%, indicating its exceptional ability to differentiate between classes with high confidence, which is crucial for minimizing false positives and false negatives in intrusion detection scenarios. This comprehensive evaluation positions ELSM favorably, demonstrating its efficacy in maintaining high accuracy while optimizing precision, recall, F1 score, and ROC AUC. The visual representation in the figure provides a clear comparative analysis, illustrating ELSM's strong performance across these critical metrics and substantiating its suitability for deployment in real-world applications that require reliable and efficient intrusion detection systems.

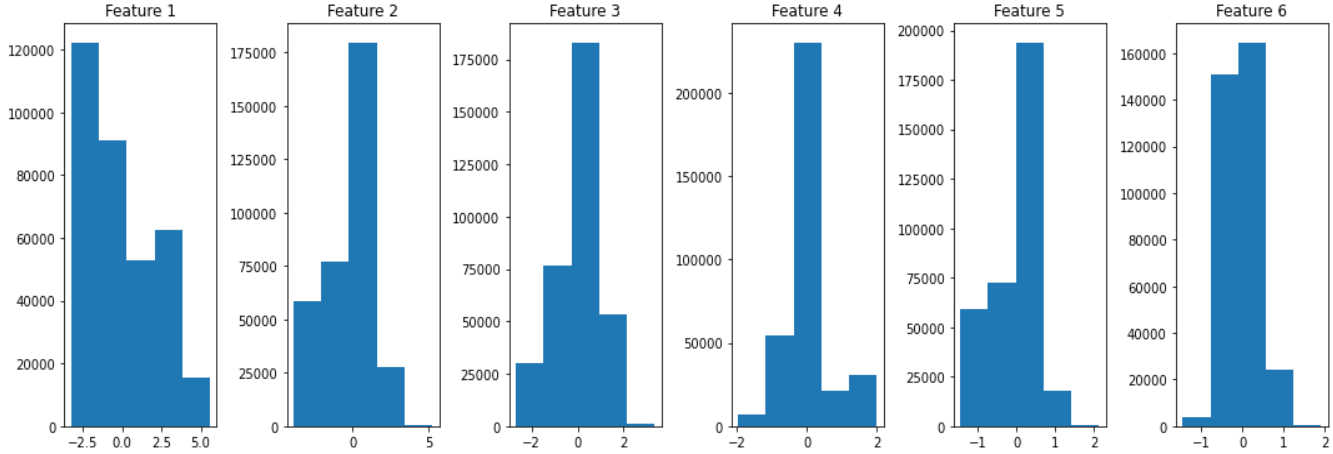

Fig. 4. The selected features with average importance above 0.0556.

TABLE II. CLASSIFIERS RESULTS AND WEIGHT

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	ROC AUC (%)	Training time (Sec)	Prediction time (Sec)	Weight	Normalized Weight
Logistic Regression	99.99	99.07	94.01	95.59	99.57	18.91	1.88e-7	5295085.502	0.3552
Decision Tree	99.97	95.06	96.78	95.82	98.39	6.77	11.16e-7	8589122.0416	0.5762
SVM	99.98	99.51	93.9	95.77	100	223.49	0.00011	9100.85	0.0006
Random Forest	99.99	99.51	91.24	92.87	99.79	114.69	8.36e-5	119585.259	0.0080
K-Nearest Neighbors	99.97	94.97	93.93	94.31	96.97	0.087	0.00018	5446.28	0.0003
XGBoost	99.99	99.51	94.01	95.83	100	6.54	1.13e-6	887210.96	0.0595
Proposed ELSM	99.97	94.97	93.93	94.31	100				

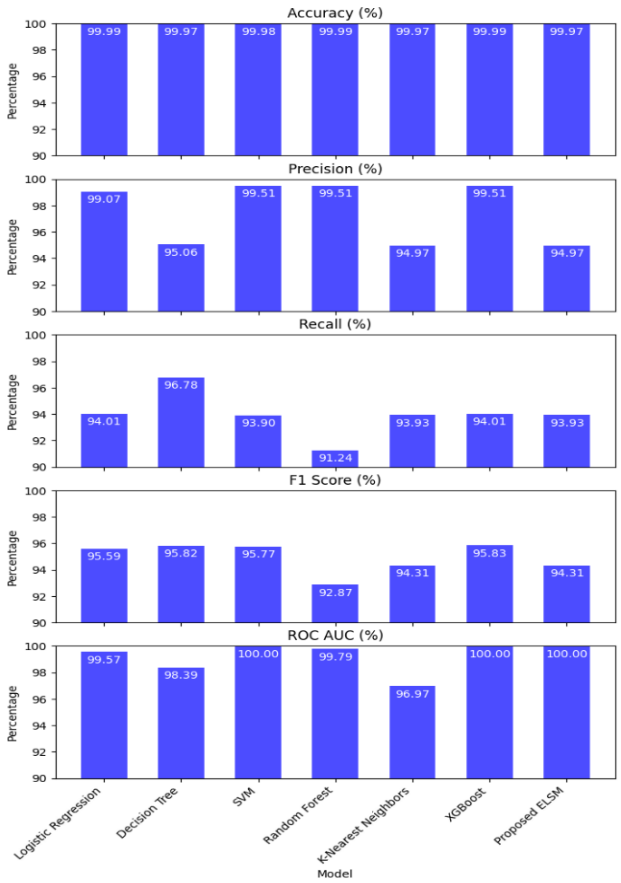

Fig. 5. Evaluation metrics for different models.

By combining these individual models using a weighted voting mechanism, the ensemble model ELSM achieved an overall accuracy of 99.97%. This high accuracy was attained by leveraging the complementary strengths of different classifiers, thereby enhancing the robustness of predictions and ensuring more reliable anomaly detection. In contrast, other methods achieved lower accuracy for the same dataset, as shown in Table I. This detailed comparison will effectively demonstrate the superiority of our approach in improving anomaly detection performance and reliability.

Subsequently, all previous stage results were integrated into the sleep-wake cycle management. The ensemble model's anomaly score determines whether the device remains in sleep mode or wakes up to handle potential threats. If the anomaly score meets or exceeds a predefined threshold, the device wakes up and sends a warning. If no anomalies are detected, the device remains in sleep mode to conserve energy. This strategic activation ensures that IoT devices are active only when necessary, optimizing power consumption while maintaining high-security standards. This approach effectively addresses the dual challenges of maintaining high-security standards and reducing power consumption, making it ideal for the resource-constrained nature of IoT environments.

V. Future Work and Challenges

Our research presents a promising approach to enhancing IoT security through ensemble learning and a strategic sleep-wake mechanism. However, several areas require further exploration. Future work will focus on scaling the anomaly detection algorithms to handle the increasing volume of IoT data while ensuring real-time processing capabilities. Additionally, the aim is to develop adaptive algorithms capable of dynamically adjusting to diverse IoT environments and data types, explore advanced feature selection techniques, and integrate the model with an edge computing framework to reduce latency.

Despite the improvements demonstrated, several challenges remain. Balancing energy conservation with high performance, addressing data heterogeneity, achieving real-time detection without compromising accuracy, and keeping up with the dynamic threat landscape are significant hurdles.

VI. Conclusion

This paper introduces a novel hybrid model designed to enhance anomaly detection in IoT environments by leveraging ensemble learning and a strategic sleep-wake mechanism, significantly improving the efficiency and effectiveness of Intrusion Detection Systems (IDS) in these settings. Our approach addresses the critical challenges of maintaining high-security standards and reducing power consumption, which are essential for the resource-constrained nature of IoT devices. The integration of multiple machine learning models using ensemble learning techniques significantly improved the overall prediction accuracy.

The experimental results demonstrated that our model achieved an impressive accuracy of 99.97% on the IoT Botnets Attack Detection Dataset. The use of feature importance assessment through Random Forest allowed us to reduce the dataset's dimensionality, focusing on the most relevant features and optimizing the model's performance and computational efficiency. By employing a weighted voting mechanism, the strengths of individual classifiers were effectively combined, enhancing the robustness and reliability of anomaly detection. Additionally, the integration of the sleep-wake mechanism ensured that IoT devices remained in a low-power state until an anomaly was detected, thereby conserving energy.

References

[1] H. Allioui and Y. Mourdi, "Exploring the full potentials of IoT for better financial growth and stability: A comprehensive survey," *Sensors*, vol. 23, no. 19, pp. 8015, 2023.

[2] N. Najari, S. Berlemont, G. Lefebvre, S. Duffner, and C. Garcia, "Radon: Robust autoencoder for unsupervised anomaly detection," in *Proc. 2021 14th Int. Conf. Security of Information and Networks (SIN)*, vol. 1, pp. 1-8. IEEE. Dec. 2021.

[3] I. S. Essop, J. C. Ribeiro, M. Papaioannou, G. Zachos, G. Mantas, and J. Rodriguez, "Generating datasets for anomaly-based intrusion detection systems in IoT and industrial IoT networks," *Sensors*, vol. 21, no. 4, pp. 1528, 2021.

[4] G. Zachos, G. Mantas, I. S. Essop, K. Porfyrakis, J. C. Ribeiro, and J. Rodriguez, "Prototyping an anomaly-based intrusion detection system for Internet of Medical Things Networks," in *Proc. 2022 IEEE 27th Int. Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pp. 179-183, Nov. 2022.

[5] Z. Shu, J. Wan, D. Li, J. Lin, A. V. Vasilakos, and M. Imran, "Security in software-defined networking: Threats and countermeasures," *Mobile Networks and Applications*, vol. 21, pp. 764-776, 2016.

[6] O. E. Tayfour and M. N. Marsono, "Collaborative detection and mitigation of DDoS in software-defined networks," *The Journal of Supercomputing*, vol. 77, no. 11, pp. 13166-13190, 2021.

[7] T. A. Tang, D. McLernon, L. Mhamdi, S. A. R. Zaidi, and M. Ghogho, "Intrusion detection in SDN-based networks: Deep recurrent neural network approach," in *Deep Learning Applications for Cyber Security*, pp. 175-195, 2019.

[8] H. Y. Ibrahim, P. M. Ismael, A. A. Albabawat, and A. B. Al-Khalil, "A secure mechanism to prevent ARP spoofing and ARP broadcasting in SDN," in *Proc. 2020 Int. Conf. Computer Science and Software Engineering (CSASE)*, pp. 13-19, Apr. 2020.

[9] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14-76, Jan. 2014.

[10] M. Said Elsayed, N. A. Le-Khac, S. Dev, and A. D. Jurcut, "Network anomaly detection using LSTM based autoencoder," in *Proc. 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, pp. 37-45, Nov. 2020.

[11] J. C. S. Sicato, S. K. Singh, S. Rathore, and J. H. Park, "A comprehensive analysis of intrusion detection system for IoT environment," *Journal of Information Processing Systems*, vol. 16, no. 4, pp. 975-990, 2020.

[12] M. Latah and L. Toker, "An efficient flow-based multi-level hybrid intrusion detection system for software-defined networks," *CCF Transactions on Networking*, vol. 3, no. 3, pp. 261-271, 2020.

[13] P. Dini, A. Elhanashi, A. Begni, S. Saponara, Q. Zheng, and K. Gasmi, "Overview on Intrusion Detection Systems Design Exploiting Machine Learning for Networking Cybersecurity," *Applied Sciences*, vol. 13, no. 13, pp. 7507, 2023.

[14] X. W. Wu, Y. Cao, and R. Dankwa, "Accuracy vs Efficiency: Machine Learning Enabled Anomaly Detection on the Internet of Things," in *Proc. 2022 IEEE Int. Conf. Internet of Things and Intelligence Systems (IoTaIS)*, pp. 245-251, Nov. 2022.

[15] Q. R. S. Fitni and K. Ramli, "Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems," in *Proc. 2020 IEEE Int. Conf. Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, pp. 118-124, Jul. 2020.

[16] A. Kumari and A. K. Mehta, "A hybrid intrusion detection system based on decision tree and support vector machine," in *Proc. 2020 IEEE 5th Int. Conf. Computing Communication and Automation (ICCCA)*, pp. 396-400, Oct. 2020.

[17] P. Pokharel, R. Pokhrel, and S. Sigdel, "Intrusion detection system based on hybrid classifier and user profile enhancement techniques," in *Proc. 2020 Int. Workshop on Big Data and Information Security (IWBIS)*, pp. 137-144, Oct. 2020.

[18] I. F. Kilincer, F. Ertam, and A. Sengur, "Machine learning methods for cyber security intrusion detection: Datasets and comparative study," *Computer Networks*, vol. 188, p. 107840, 2021.

[19] S. U. Jan, S. Ahmed, V. Shakhov, and I. Koo, "Toward a lightweight intrusion detection system for the Internet of Things," *IEEE Access*, vol. 7, pp. 42450-42471, 2019.

[20] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, "Machine learning-based IoT-botnet attack detection with sequential architecture," *Sensors*, vol. 20, no. 16, pp. 4372, 2020.

[21] A. Alzaqebah, I. Aljarah, O. Al-Kadi, and R. Damaševičius, "A modified grey wolf optimization algorithm for an intrusion detection system," *Mathematics*, vol. 10, no. 6, p. 999, 2022.

[22] A. Ugendhar, B. Illuri, S. R. Vulapula, M. Radha, K. S., F. Alenezi, et al., "A Novel Intelligent-Based Intrusion Detection System Approach Using Deep Multilayer Classification," *Mathematical Problems in Engineering*, vol. 2022, no. 1, p. 8030510, 2022.

[23] M. Otair, O. T. Ibrahim, L. Abualigah, M. Altalhi, and P. Sumari, "An enhanced grey wolf optimizer based particle swarm optimizer for intrusion detection system in wireless sensor networks," *Wireless Networks*, vol. 28, pp. 721-744, 2022.

[24] J. K. Samriya and N. Kumar, "A novel intrusion detection system using hybrid clustering-optimization approach in cloud computing," *Materials Today: Proceedings*, vol. 2, no. 1, pp. 23-54, 2020.

[25] M. Mohy-Eddine, A. Guezzaz, S. Benkirane, M. Azrour, and Y. Farhaoui, "An ensemble learning based intrusion detection model for industrial IoT security," *Big Data Mining and Analytics*, vol. 6, no. 3, pp. 273-287, 2023.

[26] E. Alalwany and I. Mahgoub, "An Effective Ensemble Learning-Based Real-Time Intrusion Detection Scheme for an In-Vehicle Network," *Electronics*, vol. 13, no. 5, p. 919, 2024.

[27] D. Jayalatchumy, R. Ramalingam, A. Balakrishnan, M. Safran, and S. Alfarhood, "Improved Crow Search-based Feature Selection and Ensemble Learning for IoT Intrusion Detection," *IEEE Access*, vol. 12, p. 32554, 2024.

[28] R. Chinnasamy, M. Subramanian, and N. Sengupta, "Devising Network Intrusion Detection System for Smart City with an Ensemble of Optimization and Deep Learning Techniques," in *Proc. 2023 Int. Conf. Modeling & E-Information Research, Artificial Learning and Digital Applications (ICMERALDA)*, pp. 179-184, Nov. 2023.

[29] S. Yangui, "A panorama of cloud platforms for IoT applications across industries," *Sensors*, vol. 20, no. 9, p. 2701, 2020.

[30] A. Yahyaoui, T. Abdellatif, S. Yangui, and R. Attia, "READ-IoT: Reliable event and anomaly detection framework for the Internet of Things," *IEEE Access*, vol. 9, pp. 24168-24186, 2021.

[31] M. S. Elsayed, N. A. Le-Khac, and A. D. Jurcut, "InSDN: A novel SDN intrusion dataset," *IEEE Access*, vol. 8, pp. 165263-165284, 2020.

[32] K. Harahsheh, R. Al-Naimat, and C. H. Chen, "Using Feature Selection Enhancement to Evaluate Attack Detection in the Internet of Things Environment," *Electronics*, vol. 13, no. 9, p. 1678, 2024.