sensors

Article

An Edge-Computing-Based Integrated Framework for Network Traffic Analysis and Intrusion Detection to Enhance Cyber–Physical System Security in Industrial IoT

Tamara Zhukabayeva [1,2], Zulfiqar Ahmad [3,*], Aigul Adamova [1,2], Nurdaulet Karabayev [1,*] and Assel Abdildayeva [1,4]

[1] Department of Information Systems, L.N. Gumilyov Eurasian National University, Astana 010000, Kazakhstan; zhukabayeva_tk@enu.kz (T.Z.); aigul.adamova@astanait.edu.kz (A.A.); assel.abdildaeva@kaznu.edu.kz (A.A.)
[2] Department of Computer Engineering, Astana IT University, Astana 010000, Kazakhstan
[3] Department of Computer Science and Information Technology, Hazara University, Mansehra 21300, Pakistan
[4] Department of Artificial Intelligence and Big Data, Al-Farabi Kazakh National University, Almaty 050040, Kazakhstan
* Correspondence: zulfiqarahmad@hu.edu.pk (Z.A.); 020419501012@enu.kz (N.K.)

Abstract: Industrial Internet of things (IIoT) environments need to implement reliable security measures because of the growth in network traffic and overall connectivity. Accordingly, this work provides the architecture of network traffic analysis and the detection of intrusions in a network with the help of edge computing and using machine-learning methods. The study uses k-means and DBSCAN techniques to examine the flow of traffic in a network and to discover several groups of behavior and possible anomalies. An assessment of the two clustering methods shows that K-means achieves a silhouette score of 0.612, while DBSCAN achieves 0.473. For intrusion detection, k-nearest neighbors (KNN), random forest (RF), and logistic regression (LR) were used and evaluated. The analysis revealed that both KNN and RF yielded seamless results in terms of precision, recall, and F1 score, close to the maximum possible value of 1.00, as demonstrated by both ROC and precision–recall curves. Accuracy matrices show that RF had better precision and recall for both benign and attacks, while KNN and LR had good detection with slight fluctuations. With the integration of edge computing, the framework is improved by real-time data processing, which means a lower latency of the security system. This work enriches the knowledge of the IIOT by offering a detailed solution to the issue of cybersecurity in IoT systems, based on well-grounded performance assessments and the right implementation of current technologies. The results thus support the effectiveness of the proposed framework to improve security and provide tangible improvements over current approaches by identifying potential threats within a network.

Keywords: network analysis; clustering; cybersecurity; intrusion detection and prevention; machine learning; cyber–physical systems; industrial IoT

check for updates

Academic Editors: Shareeful Islam, Sotiris Ioannidis, Spyros Papastergiou, Marc-Oliver Pahl and Chung Horng Lung

Received: 8 February 2025
Revised: 16 March 2025
Accepted: 27 March 2025
Published: 10 April 2025

Citation: Zhukabayeva, T.; Ahmad, Z.; Adamova, A.; Karabayev, N.; Abdildayeva, A. An Edge-Computing-Based Integrated Framework for Network Traffic Analysis and Intrusion Detection to Enhance Cyber–Physical System Security in Industrial IoT. *Sensors* **2025**, *25*, 2395. https://doi.org/10.3390/s25082395

1. Introduction

The Industrial Internet of things (IIoT) is a revolution in industrial systems that includes sensor technologies, data, and connectivity [1,2]. The IIoT uses a network of connected intelligent devices and systems that work in real time to enhance industrial processes. Integrating sensors and actuators into the machinery and equipment of an industry, IIoT systems gather large volumes of data on the machinery performance, the surrounding environment, and industry parameters [3–5]. This information is then converted into value-added outcomes by employing complex analysis and learning models to help decision-makers and increase organizational effectiveness. The basic advantages of the IIoT are mostly reflected in the concepts of automation, maintenance prediction, and operation insight. In a similar way, the IIoT helps organizations to use real-time monitoring and real-time analytics for anticipative maintenance approaches that can predict the failure of the used equipment before such a failure happens, which, in turn, reduces both the time that the equipment is not in use and the amount of money spent on maintenance [6]. Moreover, the IIoT has made operational visibility possible and gives an all-round view of the operations, making it easier to decide what next step should be taken. Consequently, the IIoT leads to critical advancements in the efficiency, protection, and utilization of resources in many industries, such as manufacturing, energy, transport, and farming industries. The IIoT is the integration of different smart devices and data analysis to redesign the future of industrial processes and create a brand-new generation of more effective and more robust industrial environments [6–8].

The security of cyber–physical systems (CPSs) in the IIoT is an important issue because it might become widely integrated into various industrial processes soon [9–11]. CPSs in IIoT applications interlink tangible equipment as well as mechanisms with intangible systems, which facilitate information sharing and command execution [12]. But this integration also presents huge security issues that need to be solved to counter various threats and risks [13,14]. The primary security concern is unauthorized access and control, as it creates an obvious information security threat. Several risk factors that affect the IIoT deployments include hacking and malware; they result in the potential compromise of the IIoT system due to the susceptibility to interferences, interruptions, or even damaging impacts on machinery data. Strengthening and enhancing authentication and authorization frameworks remains critical in safeguarding industrial systems from cyber threats [15–18]. Another important challenge is the issue of data integrity and data security. IIoT systems collect a lot of data; they include operational data, and maintenance data, as well as control data [18–20]. Such information must be appropriately transferred and archived in a manner that would not allow third-party intrusion. The use of ensembles and safe channels is necessary when it comes to protecting data from cybercriminals [7]. The nature and variety of IIoT systems and applications make it difficult to achieve coherent security requirements and policies. Having this many gadgets, protocols, and software platforms makes the IIoT systems heterogeneous and, hence, makes it challenging to implement uniform security measures. Measures such as system updates, vulnerability management, and IDS implementation should be taken to address these challenges. Protection against cyberattacks is of great importance concerning the sustainability and availability of IIoT systems and networks. Every business is prone to security incidents, and strong incident response protocols and disaster management strategies reduce the consequences and disruption of the industrial processes should they be attacked [21,22].

Edge computing is an important part of the IIoT since it optimizes the performance, dependability, and quickness of industrial processes [9,23]. It involves working with information nearer to where it is created; for instance, in the sensors and machines instead of going to massive cloud information hubs [24,25]. IIoT ecosystems produce huge amounts of data from sensors and other devices in the system. Relaying all this information to a central cloud may cause network bandwidth issues and, subsequently, high latency. Edge computing relieves this problem by processing and filtering a lot of data at the edge without sending raw data to the cloud. This maximizes the utilization of bandwidth and puts less strain on the supporting structures in the network. Edge computing enables the expansion of IIoT systems by the incorporation of more edge devices and sensors at the edge without overloading the central cloud. Such decentralization also opens more opportunities for more efficient and adaptable industrial setups, as more processing and storage resources can be deployed at different locations [9,26–28].

Security is one of the main issues of the IIoT, where the protection of cyber–physical systems becomes challenging due to the complexity and connectivity [29]. Since IIoT systems pull together increased numbers of sensors, devices, and communication networks, the exposure to cyber threats also increases, and, thus, it calls for effective security. The more complex these networks become and the faster they grow, the harder it is for traditional security solutions to protect these systems, which is why new approaches are needed in this sector. The advanced usage of edge computing can provide a solution to these security issues. Edge computing means that the data processing is performed nearer to the source, which involves IoT devices and sensors, unlike depending on a centralized cloud platform [30]. This approach not only saves time and bandwidth but also increases the ability to analyze information and make decisions in real time. If edge computing nodes are integrated into the IIoT network, traffic can be analyzed, and, potentially, threats are detected in real time, which makes the response time faster [31–33]. Another aspect addressed in improving security in IIoT systems is based on including extended traffic analysis and intrusion detection methods [2]. Clustering traffic data can be applied to analyze normal and even detect abnormal traffic flow by collecting similar datasets, which is beneficial for filtering out legitimate and illegitimate traffic. When coupled with IDSs such as k-nearest neighbors (KNN) [34–36], random forest (RF) [34,37,38], and logistic regression (LR) [34,36,39,40], potential intrusions can be identified and classified correctly using the observed traffic pattern.

This research is driven by the growing importance and critical nature of cybersecurity in industrial IoT systems. With growing numbers of industries implementing IoT into their processes, the intensity and variety of network load remain high, which poses enormous problems for the creation of effective security systems. Industrial IoT systems are embedded in many industries to facilitate the control and monitoring of critical processes in manufacturing and energy, among other industries in transportation. In such environments, the potential of cyber threats is vast, thus implying that, even though protective measures are important, they are sensitive. Traditional approaches often struggle to adapt to the diverse and evolving traffic patterns in industrial IoT networks, and this makes them less effective against sophisticated cyberattack strategies [13,16,41–44]. Moreover, this work is inspired by the desire to move beyond traditional security solutions and come up with better, more flexible approaches to network analysis and intrusion detection.

The main contributions of the proposed research study are highlighted below:

- We propose a comprehensive framework that integrates network traffic analysis, clustering, intrusion detection, and edge computing to address the complexity and security challenges of cyber–physical system security in industrial IoT.
- We employ k-means and DBSCAN clustering methods to effectively segment network traffic and, therefore, detect different traffic scenarios in IIoT networks as well as potential discrepancies.
- We implement and compare three machine-learning models (KNN, RF, and LR) to detect intrusions and found the models performing well with a high accuracy to differentiate between benign and malicious traffic.
- We provide the practical mechanism of incorporating edge computing within the proposed framework.
- We provide recommendations regarding clustering and machine-learning approaches that are suitable for practical implementation in actual industrial IoT environments.

We organized the remaining part of the paper as follows: Section 2 describes the related work. Section 3 employs the system design and model. Section 4 presents a performance evaluation, and, finally, Section 5 concludes the article with several future directions.

2. Related Work

We reviewed the related work in context with security issues in cyber–physical systems, edge computing integration, network analysis, and intrusion detection methods for cyber–physical systems.

With the emergence of cyber–physical systems, people confronted new opportunities. Protecting information from cyber–physical systems is one of the most challenging questions in a vast number of protections against cyber threats. The purpose of the study presented in [16] is to review the literature related to the security of cyber–physical systems and categorize them. The nature of cyber–physical systems is briefly described in terms of philosophical problems. The authors also identified the main types of attacks and threats against cyber–physical. In the recent past, the research on the CPS has been receiving much attention in the research community as well as in industries [27]. The first challenge to the rapid deployment of CPS applications is the lack of an idea on how best to deal with the large volumes of data that the applications generate for decision-making. In response to this challenge, scholars propose the concept of incorporating edge computing, or edge–cloud computing, into the design of CPSs. However, this coupling process brings a diversity of issues in the QoS of CPS applications into question. In [27], the authors offer an overview of the edge-computing- or edge–cloud-computing-assisted CPS designs from the QoS optimization point of view. The authors provide a brief of the state-of-the art works addressing divergent issues related to QoS optimization.

The transformation of the fourth industrial revolution uses a CPS that protects Supply Chain 4.0. It combines manufacturing information with Internet communication technology to design a smart CPS that captures product data from manufacture to customer delivery by the Internet of things. The research presented in [13] employs a ML method for network anomaly detection and building models to identify DDoS attacks targeting Industry 4.0 CPSs. The limitations of the previous techniques, artificial data, and small datasets are eliminated by collecting real-world network traffic data from a semiconductor production factory. The presented PCA-BSO algorithm is used to identify the most significant features by using their eigenvalues, though the feature that has the largest eigenvalues is not necessarily beneficial to improving the classification. As for the supervised ML algorithms, the simulations are performed to compare the results of the algorithms. Hackers consistently create new approaches on how to confuse and deceive victims, making cybersecurity an ongoing process aimed at maintaining the availability, confidentiality, and integrity of computer systems [41]. Proactive protection is engaged by employing machine learning (ML) as an effective approach to the intelligent cyber analysis of recurring patterns of successful attacks. However, two significant drawbacks hinder the widespread adoption of ML in security analysis: relatively high computing overheads and the requirement of specialized frameworks. The study presented in [41] is intended to establish the numerical value by which a hub can improve the safety of an ecosystem. Traditional cyberattacks were performed on an IoT network in a smart house to test the functionality of the hub. The robustness of the IDS against adversarial machine-learning (AML) attacks was explored, in which models are attacked with adversarial samples to exploit vulnerabilities.

With advanced technology, most devices have been created to share extensive information and work in unison as Edge Intelligence in Smart Cities (EISC). In circumstances where personal data are processed, great care must be taken to make sure that personal data are not disclosed, and there should not be any disclosure of any information. Systems such as IDSs, are needed to penetrate firewalls, anti-virus programs, and other protective equipment to ensure total system protection in smart operating systems. There are three aspects to an IDS: the intrusion detection method, the architecture, and the intrusion response method. In [42], linear correlation feature selection methods and cross-information are used in combination. The dataset used in the article is KDD99. The paper, therefore, discusses using two approaches in attack prediction in intrusion detection systems known as INTERACT and a multilayer perceptron (MLP). Since the record count for each attack type may not be the same, one of the recommendations is to keep on using data-balancing approaches.

A comparison of the proposed framework with previous studies is shown in Table 1, which highlights the ML techniques used, the features considered, and the datasets utilized. The traditional approaches struggle to adapt to the diverse and evolving traffic patterns in industrial IoT networks and are less effective against advanced cyberattack strategies as compared to the proposed approach.

Table 1. A comparison of the proposed framework with previous studies.

Reference	Focused Area	Machine Learning Techniques	Features	Dataset	Key Contributions
[13]	Anomaly detection in CPS for Supply Chain 4.0	PCA-BSO feature selection + ML models	Network traffic anomalies	Real-world traffic data from a semiconductor factory	PCA-BSO used to select significant features, improving classification accuracy
[16]	Security issues in cyber–physical systems (CPSs)	Review-based work	CPS security threats and attack types	Review-based work	Categorizes CPS security threats and describes philosophical challenges
[27]	Edge computing integration in CPS	Review-based work	QoS optimization in CPS	Review-based work	Edge–cloud computing and QoS challenges in CPSs
[41]	Machine-learning-based cybersecurity analysis	Network IDS and Host IDS	Attack pattern analysis, adversarial ML vulnerabilities	KDDCup99, NSL-KDD, UNSW-NB15, WSN-DS, CICIDS 2017	Explores robustness of IDSs against adversarial ML attacks
[42]	IDS in Edge Intelligence for Smart Cities	Linear Correlation Feature Selection, INTERACT, MLP	Feature extraction and selection for attack prediction	KDDCup99	Hybrid feature selection and ML for improved intrusion detection
Proposed Framework	Network traffic analysis and intrusion detection in IIoT CPS using edge computing	k-means, DBSCAN, KNN, RF, LR	Network traffic behavior, anomaly detection, supervised learning for intrusion detection	NF-ToN-IoT-V2 (real-world industrial IoT dataset)	Integrates clustering and ML models with edge computing for real-time and scalable intrusion detection

3. System Design and Model

We propose an edge-computing-based integrated framework for network traffic analysis and intrusion detection to improve CPS security in industrial IoT, as illustrated in Figure 1. The framework is intended for use in the IIoT for CPSs, where traffic is constantly analyzed for security threats. The data used in the study are collected from a CPS within the IoT environment; the CPS collects different types of network traffic data. These traffic data also consist of normal and intrusion instances. Raw network traffic undergoes preprocessing before analysis. Preprocessing encompasses data cleansing, and scaling, as well as data transformation and dimensionality reduction.

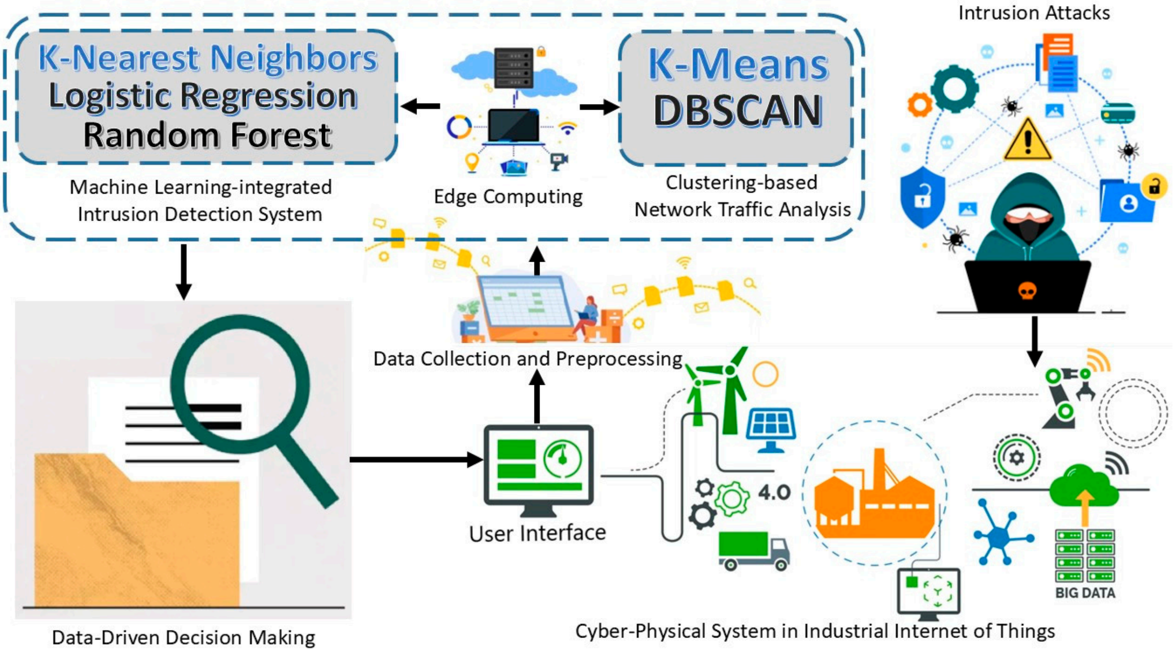
K-Nearest Neighbors
Logistic Regression
Random Forest
Machine Learning-integrated Intrusion Detection System
Edge Computing
K-Means
DBSCAN
Clustering-based Network Traffic Analysis
Intrusion Attacks
Data Collection and Preprocessing
User Interface
4.0
BIG DATA
Data-Driven Decision Making
Cyber-Physical System in Industrial Internet of Things

Figure 1. An edge-computing-based integrated framework for network traffic analysis and intrusion detection to improve CPS security in industrial IoT.

Two clustering algorithms, namely, k-means and DBSCAN, are used to analyze the network traffic data in an unsupervised manner. The purpose is to find specific groups of points that characterize other traffic behaviors or the presence of outliers, which can signal cyberattacks. The algorithm, k-means clustering, divides the data into predetermined clusters, implying well-developed clusters. Density-based spatial clustering of applications with noise (DBSCAN) finds clusters and noise with different densities in clusters. The preprocessed data are also used in evaluating the performance of three supervised machine-learning models for intrusion detection. The basic and efficient KNN algorithm in this study has allowed us to gain good performance in terms of precision, recall, and F1 scores. The better precision and recall of the RF ensemble-learning method were also achieved when used solely for distinguishing both benign and attack traffic, particularly in several attack classes. LR has the ability to show good detection performance in intrusion detection, and, therefore, it was also used in the proposed framework. An important part of the given framework is the incorporation of edge computing. Since the analysis of the network traffic data is performed at the network periphery, the proposed framework prevents the high latency that is inherent to most existing solutions, as well as enabling real-time intrusion detection. This has the effect of improving the system's capability to rapidly identify antecedents and respond to dangerous elements. This is followed by the network traffic analysis and intrusion detection, after which the system uses a data-based decision-making function to decide on the best course of action to be taken in case of any security threat. In the occurrence of recognized anomalies or intrusions, the system can perform such actions as alerting security staff, disconnecting the infected devices, or automatically starting defense procedures.

The elevation of security measures in the cyber–physical system of the IIoT is confirmed to be effectively addressed by the proposed framework. By applying clustering for network data analysis and machine learning for intrusion detection with edge computing for real-time operation, the proposed approach guarantees the swift and precise detection of security threats. The system helps improve cybersecurity by providing the capability for a faster response and better decision-making to protect the infrastructure from industrial IoT.

3.1. Implementation of Edge Computing in the Proposed Framework

By integrating edge computing in the proposed framework, it enhances the performance on network traffic analysis and intrusion detection by the timely processing of the data at the edge of the network. As shown in Figure 2, the distributed task computing is carried out at the edge layer instead of depending on centralized cloud servers alone, which distributes the computation over multiple edge devices, including routers, gateways, and fog nodes. There are these intermediate processing units, which are edge nodes, which analyze the network traffic closer to the data source and, thus, reduce the latency and response time for security threats in the IIoT environments. The framework uses edge computing to make sure that critical security operations such as clustering and machine-learning-based intrusion detection as well as an automated threat response are carried out efficiently without burdening the central cloud infrastructure.

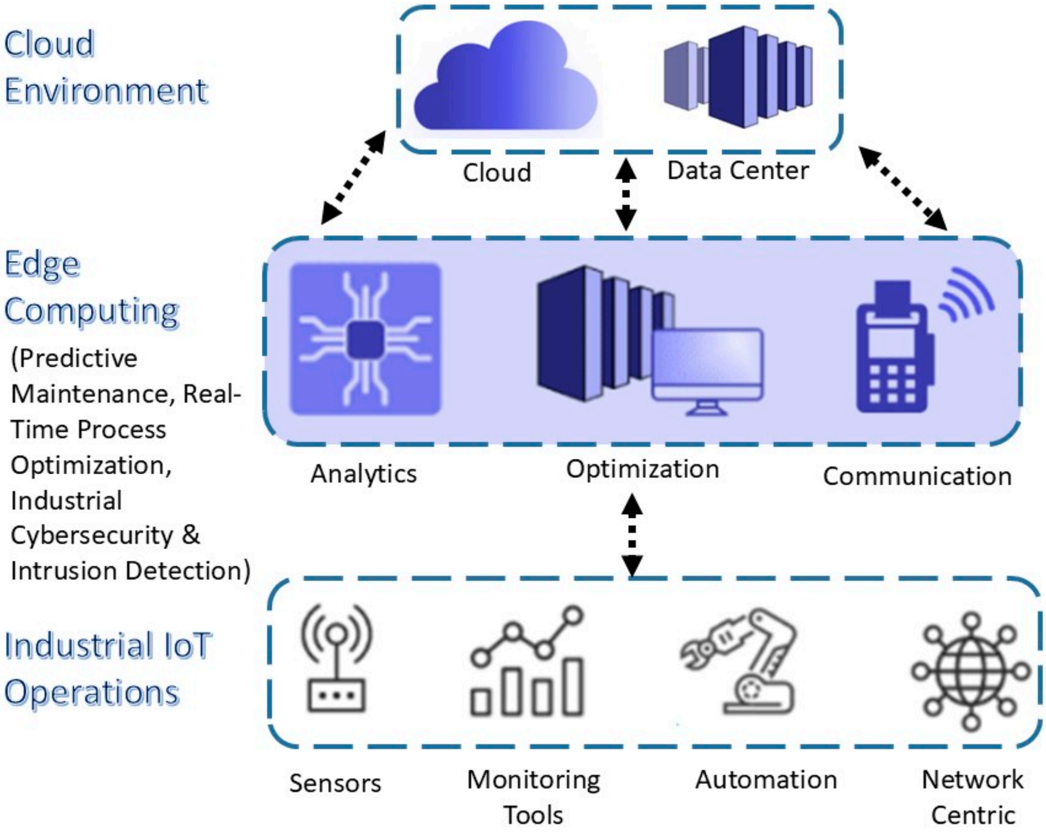
Cloud Environment
Cloud
Data Center
Edge Computing
(Predictive Maintenance, Real-Time Process Optimization, Industrial Cybersecurity & Intrusion Detection)
Analytics
Optimization
Communication
Industrial IoT Operations
Sensors
Monitoring Tools
Automation
Network Centric

Figure 2. Edge computing implementation in IIoT.

Real-time data collection and preprocessing at the edge is the first step towards the implementation of edge computing. As IIoT environments have lots of CPSs, the network traffic generated from these CPSs is large and includes normal and anomalous data packets. All network traffic would be routed to a cloud-based security system for analysis. In the proposed framework, however, data are captured and preprocessed at edge-computing devices, and only the relevant data are sent to the cloud. The first part of edge computing is data cleansing, feature extraction, normalization, and dimensionality reduction, which are carried out as preprocessing tasks at the edge. These operations discard redundant or noisy data and only pass on useful and security-relevant data for further processing. The reduction in network congestion and maximization of bandwidth utilization are significantly achieved as the amount of raw data transmitted to central servers is minimized.

With the data preprocessed, it is fed at the edge and analyzed using clustering techniques such as k-means and DBSCAN. The clustering algorithms are executed on the edge nodes to segment network traffic into different behavioral patterns for the purpose of the early detection of anomalies that can be indicative of potential cyber threats. K-means clustering is highly effective for clustering structured traffic patterns so that normal and suspicious activities can be identified with high precision. However, DBSCAN is useful for detecting irregular, sparsely distributed, and noisy anomalies and, thus, is very useful in detecting outliers and noise in network traffic. By performing clustering at the edge, the system can quickly identify potential security breaches and forward only the high-risk data instances to the central security server for further investigation. On top of this, the framework deploys machine-learning intrusion detection models at the edge for security-monitoring enhancement. In order to classify network traffic in real time, we implement three machine-learning algorithms on edge-computing devices using KNN, RF, and LR. The models that have been trained on historical intrusion datasets analyze the incoming network packets and decide if they are normal or attack traffic. For its initial classification step, the KNN algorithm is used as an algorithm with a simple implementation and high efficiency, while the second has the higher accuracy to classify whether the traffic is benign or malicious—an RF model, which has the robustness for distinguishing between benign and malicious traffic. The LR model helps in further improving the classification capability by modeling the probability of the occurrence of the attack. Given that early detection and mitigation are crucial, these computations can be carried out at the edge on time compared to cloud-based solutions, thereby drastically reducing the time taken for detection.

The important advantage that is brought in by the incorporation of edge computing in the proposed framework is that it helps in providing real-time decision-making as well as automated response mechanisms. In the case of an intrusion or anomaly at the edge, the system responds immediately and without waiting for any orders sent by the centralized cloud server. Actions that may be implemented by the response part of these are isolating compromised devices, blocking suspicious network traffic, updating firewall rules dynamically, or sending the alerts to the security personnel. Edge computing is decentralized, which means security incidents are dealt with at the place of origin, thereby stopping any potential threats from spreading throughout the whole IIoT network. Edge computing makes it possible to have the continuous monitoring of the network behavior and then respond with adaptive security mechanisms on the fly according to the changing nature of the cyber threats.

3.2. Algorithm for Edge-Computing-Based Integrated Framework for Network Traffic Analysis and Intrusion Detection for CPSs in IIoT

The steps of the proposed framework are shown in Algorithm 1. It provides a detailed approach to analyzing the network traffic, identifying intrusions in industrial IoT settings, and incorporating edge computing as a means of processing data in real time. It begins with the raw network traffic data collected from CPSs in the IIoT, referred to as the raw data. These data are then preprocessed through normalization, which means scaling the features in the given range between 0 and 1, making the dataset normalized. This step is important because it helps in cases where the data are going to be used repeatedly. The next phase involves network traffic analysis, in which two clustering algorithms, namely, k-means and DBSCAN, are implemented. The k-means algorithm is used for clustering data, for which minimization of the variance within clusters is the main objective. The process of clustering is achieved by assigning each data point to the correct cluster center, and the efficiency of the generated clustering is estimated through the silhouette coefficient, measuring the appropriateness of the data points belonging to a specific cluster. At the same time, DBSCAN is performed to find dense areas and assess the outliers through parameters for the size of the neighborhood and minimum points. This density-based method works in conjunction with k-means by identifying any clustering that is otherwise labeled as outliers. The silhouette score is also used to measure the level of efficiency that comes with DBSCAN clustering.

The intrusion detection process follows, utilizing three machine-learning models: k-nearest neighbors, random forest, and logistic regression. The normalized dataset divides the entire dataset into training and testing sets. K-nearest neighbors aimed to assign a class to each test instance as the majority from the nearest neighbors in a training set, random forest makes a team of decision trees; the prediction is made by a simple voting system of all the trees, while logistic regression employs a probability model for predicting the probability of intrusion using a cost function. These models present a sound way of detecting intrusions and abnormalities. To facilitate the real-time processing of data, the proposed framework incorporates edge computing, through which information is processed near the source, thus increasing efficiency. It also enhances the prompt identification and the subsequent handling of prospective threats by shifting computational tasks to the edge.

The algorithm also has decision-making based on the data collected from clustering and intrusion detection to decide the subsequent action to be taken. If there is a high probability that an anomaly or an intrusion occurred in the system, then the system can alert the administrators or even quarantine the culprits' devices. In the case where no problems are flagged, the process proceeds without changes to the pre-existing plan. The algorithm contains a performance assessment. The performances of the clustering models are measured with the silhouette score, and the performances of the intrusion detection models are evaluated by the precision, recall, F1 score, and AUC. The confusion matrices are evaluated to estimate the accuracy of each model to distinguish between normal and attack traffic and to confirm the effectiveness of the proposed framework to detect potential threats in industrial IoT environments.

Algorithm 1 Edge-Computing-Based Integrated Framework for Network Traffic Analysis and Intru-sion Detection for CPSs in IIoT

```
Begin
 Input:   $D_{RAW}$: Raw network data from the CPS in IIoT
 Output:  Cluster-based IDS
Procedure: Edge-Computing-Based Integrated Framework for Network Traffic
           Analysis and Intrusion Detection for CPSs in IIoT
 Data Collection:
    • Collect raw network traffic data from the CPS in IIoT
          $D_{RAW} = \{x_1, x_2, \dots, x_n\}$
Data Preprocessing:
    • Preprocess the raw data $D_{RAW}$ using normalization to scale features within
      [0, 1]
          $D_{norm} = x_i^{norm} = \frac{x_i - \min(x)}{\max(x) - \min(x)}, \forall\, x_i \in D_{RAW}$
Network Traffic Analysis:
    • Apply k-means clustering to $D_{norm}$
        o Minimize intra-cluster variance
```

Algorithm 1 *Cont.*

```
                $arg \underset{c}{min} \sum_{i=1}^{k} \sum_{x_j \in C_i} ||x_j - \mu_i||^2$
            where $\mu_i$ is the centroid of cluster $C_i$ and $C_i$ represents the set of points in cluster i
        o   Assign each data point $x_j$ to the closest cluster
                $Cluster\ (x_j) = arg \underset{c}{min} ||x_j - \mu_i||$
        o   Computing the silhouette score $S_{K-Means}$
                $S_{K-Means} = \frac{b(x) - \mathrm{a}(x)}{\max(a(x), b(x))}$
            where a (x) is the mean intra-cluster distance, and b (x) is the mean nearest-cluster distance
    •   Apply DBSCAN clustering to $D_{norm}$
    for each point ($x_j$)
        {
        classify core points if it has at least $MinPts$ neighbors within $E$
        classify border point if it is within $E$ of a core point but has fewer than $MinPts$ neighbors
        Outlier otherwise
        }
    end for
        o   Computing the silhouette score $S_{DBSCAN}$
8.  Intrusion Detection using Machine-Learning Models:
    •   Let $D_{Train}$ and $D_{Test}$ represent training and testing dataset derived from $D_{norm}$
    •   Apply k-nearest neighbors
        o   For each test instance $x_{test}$, find k-nearest neighbors in the training set
                $Neighbors\ (x_{test}) = arg \underset{k}{min} ||x_{test} - x_{train}||$
        o   Classify $x_{test}$ based on majority of its neighbors
    •   Apply random forest
        o   Construct T decision trees, each trained on random subset of $D_{Train}$
        o   For a test instance $x_{test}$, predict the class $y_{test}$ by averaging the predictions of all trees
                $y_{test} = mode\ (\{h_t(x_{test}) | t = 1, 2, \ldots, T\})$
    •   Apply logistic regression
        o   Train LR model by fitting the parameters $\theta$ to minimize the log-loss
                $\theta^* = arg \underset{k}{min} (-\frac{1}{m} \sum_{i=1}^{m} [y_i \log(h_\theta(x_i)) + (1 - y_i) \log(1 - h_\theta(x_i))] )$
        o   where $h_\theta(x)$ is the logistic function: $h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$
    •   Compute accuracy (A), precision (P), recall (R), and F1 score.
                $A = \frac{TP + TN}{TP + TN + FP + FN}$
```

Algorithm 1 *Cont.*

```
        $P = \frac{TP}{TP + FP}$
        $R = \frac{TP}{TP + FN}$
        $F1 - Score = \frac{2(P \times R)}{P + R}$
9.  **Edge Computing Integration and Data-Driven Decision Making:**
    •  Let $T_{Edge}$ represent the time taken for edge-based data processing and $T_{Cloud}$ represent the time for cloud-based processing
        $T_{Edge} \ll T_{Cloud}$
    •  Let A represent the set of actions triggered in response to detected intrusions
        $A = \begin{cases} Alert\ if\ P_{anomaly}(x) > T \\ Isolate\ Device\ if\ P_{attack}(x) > T_{attack} \\ Normal\ Operation\ Otherwise \end{cases}$
       where $T$ and $T_{attack}$ are thresholds for anomaly detection and attack classification
10. ***end***
```

4. Performance Evaluation

We perform simulations and evaluate the performance of the proposed framework with respect to the cluster-based network analysis and intrusion detection system.

4.1. Evaluation Metrics

We evaluated the performance of the methods and models implemented in the proposed framework using the silhouette score, accuracy, precision, recall, F1 score, receiver operating characteristics (ROC) curve, and precision–recall (PR) curve [45]. We calculated the accuracy, precision, recall, and F1 score based on the following terms:

- True Positive (TP): the number of correctly identified positive instances;
- True Negative (TN): the number of correctly identified negative instances;
- False Positive (FP): the number of incorrectly identified positive instances;
- False Negative (FN): the number of incorrectly identified negative instances.

The accuracy, precision, recall, and F1 scores are calculated based on the following equations [46,47]:

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$P = \frac{TP}{TP + FP} \tag{2}$$

$$R = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = \frac{2(P \times R)}{P + R} \tag{4}$$

The ROC and PR curves have also been used during evaluation. The ROC curve is a graphical representation of the TP rate against the FP rate at various threshold levels. The PR curve is a graphical representation of the trade-off between the precision and recall for different classification thresholds. A higher area under the curve (AUC) indicates a better performance for the model.

4.2. Dataset

We have utilized the dataset called NF-ToN-IoT-V2 [48], which is publicly available on the Kaggle platform. This dataset is obtained from the NFV2-collection developed by the University of Queensland to remove issues of interoperability in network security datasets for scalability. The dataset is an integrated dataset that strives to emulate the real IIoT scenario and contains information about IoT sensors, OS, and network traffic. It is labeled for several cybersecurity incidents, such as distributed denial of service (DDoS) and ransomware, among others. This is composed of normal traffic and attack traffic, which makes it suitable for use when developing machine-learning models for intrusion detection and any other security-related concerns. For this purpose, in this study, we employed this dataset for network traffic anomaly detection, where the labeled traffic data can be employed to categorize the anomalous patterns.

In the context of the proposed framework, the NF-ToN-IoT-V2 dataset is a better option for evaluating the effectiveness of network traffic analysis and intrusion detection models with our proposed framework. This dataset is specifically created to simulate realistic IIoT environments that were generated by different IoT sensors, the operating system, and network communications. It is normal and malicious traffic that encompasses many cybersecurity threats such as DDoS, ransomware, and brute force attacks. The dataset is scalable and has a rich feature set to capture the dynamic network conditions and, therefore, can be used to evaluate the robustness of intrusion detection mechanisms in IIoT scenarios. Since our frameworks include clustering methods (k-means and DBSCAN) and supervised learning methods (KNN, RF, and LR) for intrusion detection, our dataset should include as many of the different behaviors of IIoT traffic in terms of network load variation and anomaly patterns. For this, the real-world-inspired traffic flows are incorporated in the dataset, which leads to the adequate training and evaluation of the model with the labeled instances of cyber threats available in the dataset.

4.3. Experimental Design

The experiments were performed by implementing two clustering methods, k-means [49,50] and DBSCAN [51], and three ML models, k-nearest neighbors [34–36], random forest [34,37,38], and logistic regression [34,36,39,40]. The dataset has been divided into two parts: the training set and the test set. The training set comprised 70% of the total records in the dataset. The test set comprised 30% of the total number of records. All experiments are implemented in Python 3.11.8 in a GPU-based environment. Predefined ML packages and libraries including Pandas 1.24.0, Numpy 1.5.3, Seaborn 0.11.2, Sklearn 1.1.3, LabelEncoder (from Scikit-learn), OneHoTencoding (from Scikit-learn), and Matplotlib 3.6.3 have been used.

4.4. Results and Discussion

4.4.1. Network Traffic Analysis

A network traffic pattern analysis was conducted using the k-means clustering approach, in which the "IN_BYTES" and "OUT_BYTES" parameters were used. We then grouped the network traffic into five separate categories based on the 5000 data points collected. K-means clustering was selected as it can be used for the analysis of datasets with several variables to discover underlying trends and, thus, enable us to investigate possible traffic abnormalities or different behavior patterns in the network. The distribution of instances across the five clusters is shown in Table 2.

Table 2. Distribution of instances across the five clusters.

Cluster	Number of Instances
Cluster 4	2089
Cluster 0	1808
Cluster 1	624
Cluster 2	393
Cluster 3	86

To determine the performance of the clustering, the silhouette score was used as a measure of clustering quality in order to know how well the clusters were separated. From this, a score of 0.612 was obtained, which is good for clustering quality since the clusters are well-separated. Figure 3 shows a graphical representation of the clustering solution based on the analysis of the "IN_BYTES" and "OUT_BYTES" instances normalized for analysis. Different colors have been used to represent various clusters. The x-axis plot shows that some of the clusters are spread more widely, which may imply different traffic densities or traffic patterns, while other clusters are found to be as near to zero as possible, which could imply normal traffic or a lack of traffic in the network. This analysis is important in identifying patterns that may be indicative of anomalous traffic behavior and, hence, improve security and optimization in industrial IoT systems.

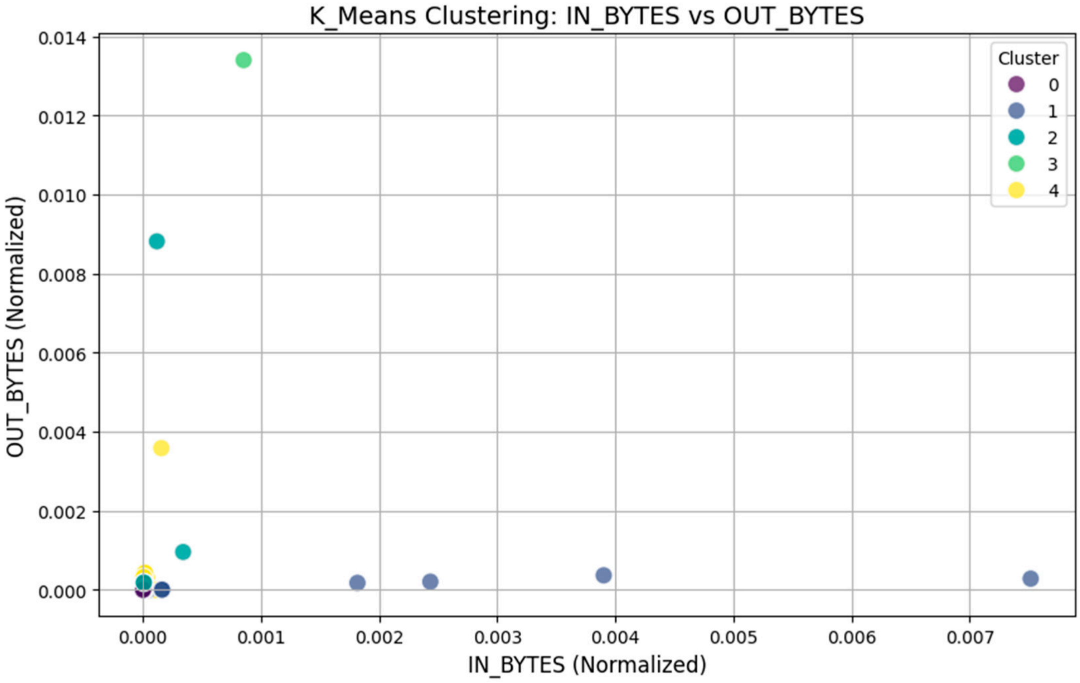

Figure 3. K-means clustering: IN_BYTES Vs. OUT_BYTES.

Apart from k-means clustering, we used DBSCAN clustering to analyze the network traffic and search for regular/abnormal traffic patterns. DBSCAN is especially useful when the clusters are irregular in shape and size and contain noise. For the current analysis, the DBSCAN algorithm was used under parameters that include epsilon (eps) of 0.1 and min_samples of 5. When analyzing the results of resource utilization, the algorithm found several clusters, each characterizing a group of similar traffic patterns in the network. Entropy for the DBSCAN clustering was also calculated and we obtained a silhouette score of 0.473. It is lower than the score obtained here by k-means; DBSCAN is more robust against noise, and its results may shed a lot of light on isolating traffic, which could be a sign of possible anomalies or security threats.

The outcomes of DBSCAN clustering are shown in Figure 4, with IN_BYTES normalized in the x-axis and OUT_BYTES normalized in the y-axis. The clusters can be uniquely colored, with different shades belonging to different groups of clusters. DBSCAN also focuses on how the data are distributed, and can easily spot regions with fewer database points, which is potentially an area of noise or outliers.

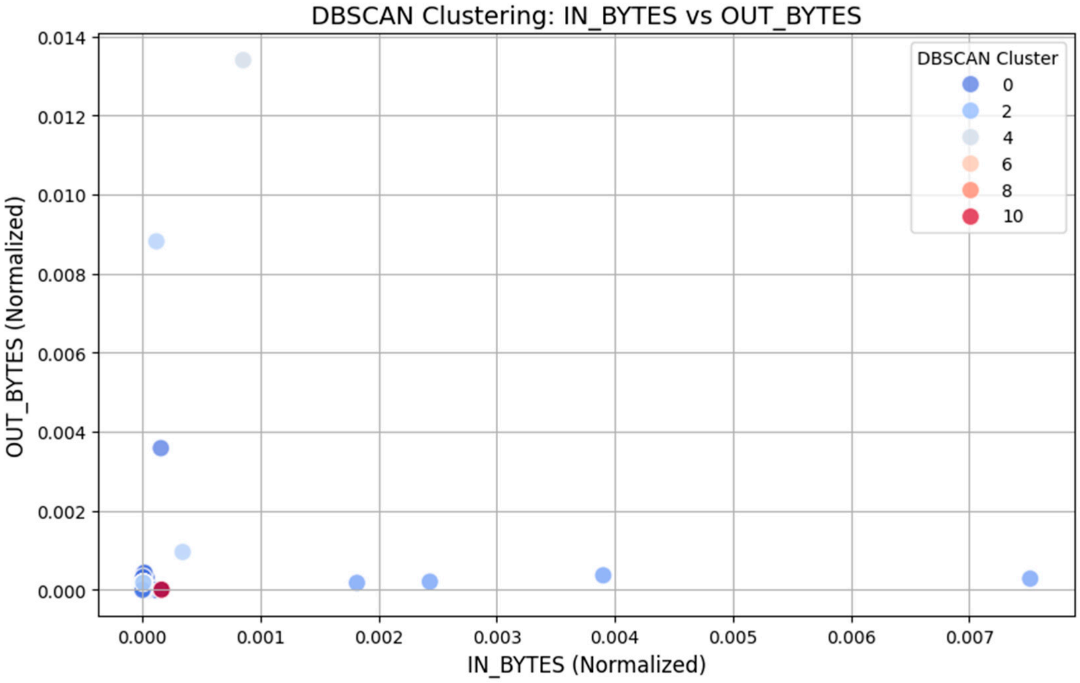

Figure 4. DBSCAN clustering: IN_BYTES Vs. OUT_BYTES.

The network traffic analysis with k-means and DBSCAN clustering algorithms is quite different in their approach. The most popular and fast-calculation algorithm, K-Means, divides the set into five clusters according to the “IN_BYTES” and “OUT_BYTES” attributes. The clustering yielded a silhouette coefficient of 0.612, which shows that the clustering resulted in a high separation of network traffic patterns. However, k-means is sensitive to the choice of k, also assumes clusters are spherical, and fails to work with noisy clusters or clusters of an irregular shape; thus, its ability to identify anomalies or outliers in datasets, especially those that are complex, may not be as effective. DBSCAN clustering, on the other hand, is density-based, efficient for discovering clusters of any shape, and effective in the presence of noise. It also did a better job in identifying the low-density areas that might be representative of abnormal or infrequent traffic patterns. Even though it is more flexible than k-means, DBSCAN achieved a silhouette coefficient of 0.473 because it is designed to identify noise in the data rather than creating compact and clearly separated clusters. That makes it especially important for detecting possible security breaches or other suspicious activities in network communication, where outliers and low-density data are important to find.

The silhouette scores quantify the level of clustering quality; however, the use of k-means and DBSCAN is not based on these scores alone but on their strengths in dealing with different IIoT network traffic patterns. Our framework is based on the use of the NF-ToN-IoT-V2 dataset, which contains heterogeneous traffic with normal operations and different attack scenarios. However, such variability suggests that k-means is useful for segmenting structured patterns efficiently, whereas DBSCAN is good at detecting anomalies and noise. In our proposed framework, we employ both of them as they help in a comprehensive analysis via k-means to identify the principal traffic clusters and DBSCAN for outliers and irregular traffic behavior. We have also investigated the stability and interpretability of clusters by considering intra-cluster variance and the effectiveness of detecting the anomalies.

4.4.2. Intrusion Detection System

We implemented and evaluated three ML models, KNN, RF, and LR. All these models have expertise in identifying network intrusions in the industrial IoT networks. KNN is a distance-based classifier, and it is good at capturing the local structures and finding the outliers in the dataset. RF, as one of the popular ensemble methods, provides a high efficiency in dealing with large datasets with the high dimensionality of features and does not allow overfitting. LR, which is a probabilistic model, is particularly useful for binary classification such as intrusion detection and can explain the relative association between features. Comparing the results of the use of these models, we identify the best strategy for detecting intrusions in real time within cyber–physical systems. The results of the IDS models, i.e., KNN, RF, and LR, are shown in Table 3 and demonstrate varying levels of performance across multiple metrics.

Table 3. Comparative analysis of accuracy, precision, recall, and F1 score.

Class	Precision			Recall			F1 Score			Accuracy		
	KNN	RF	LR	KNN	RF	LR	KNN	RF	LR	KNN	RF	LR
Benign (0)	0.99	1.00	0.96	0.98	0.98	0.89	0.99	0.99	0.92	0.99	0.99	0.97
Attack (1)	1.00	0.99	0.97	1.00	1.00	0.99	1.00	1.00	0.98			
Macro average	0.99	1.00	0.97	0.99	0.99	0.94	0.99	0.99	0.95			
Weighted average	0.99	0.99	0.97	0.99	0.99	0.97	0.99	0.99	0.97			

In the benign class, the precision of the RF was 1.00, indicating that all instances that the RF model classified as benign are indeed benign. KNN was also good with a precision value of 0.99, while the LR had a lower value of 0.96, which means that the model predicted a few false positives. On the same note, the recall of RF and KNN stood at 0.98 and that of LR was 0.89. From this, it can be inferred that LR failed to identify more benign cases, lumping them together with the attacks. RF emerged the best with an F1 score of 0.99, followed by KNN. However, the F1 score that was calculated for LR was 0.92, which indicates that the model's recall is worse than its precision. Based on accuracy, as seen above, RF and KNN produced almost similar results with an accuracy of 0.99 for benign traffic, while LR, though equally accurate, had a slightly lower value of 0.97.

In the case of the attack class, KNN and LR are both valued at 1.00 for the precision, which means that the classifying models did not misclassify any attack instances, and all the instances are correctly labeled as attacks. The RF model gave a slightly lower accuracy of 0.99, which also proved good enough. For recall, all models achieved a perfect recall of 1.00 on attack detection except for the LR model, which achieved 0.99. This means that, although LR was active, it did not miss very many attacks, which is not true according to the results. The F1 scores also supported this trend, with both KNN and RF models having the same F1 score of 1.00, but with F1 metrics being the harmonic means of precision and recall. LR has a slightly lesser F1 measure of 0.98, but it is also significantly good.

When comparing the results of the models on macro averages across both benign and attack classes, all models were good, with RF having the highest performance with scores of 1.00 in precision, recall, and F1. KNN, with an average of 0.99, showed a good trade-off between both true benign and true malicious activities. LR slightly performed worse with 0.97 precision, 0.94 recall, and 0.95 F1 score and was slightly less accurate and able to detect the benign class. In the same manner as the accuracy averages, which also take into account the number of instances in each class, RF and KKN achieved very high scores of 0.99 in all the metrics, implying a robustness in both benign and attack classes. LR had a slightly lower weighted average of 0.97 because it lowered its detection performance for benign traffic.

Random forest (RF) showed the highest accuracy for the total score and the highest specific, non-specific, and overall accuracy of the attack and benign classes. This is the reason why the performance of RF is better since it can cope with the complicated data patterns with an appropriate degree of predictability. The KNN model also provided good results, even better than the previous models, especially for attacks where all the metrics were scored 100%. But it had a slightly lower recall in the case of benign instances as compared to RF, which means that there may be a small number of benign events being classified as an attack. Although LR was good at recognizing attacks, it had deficiencies in identifying benign traffic, as depicted by its low recall and F1 scores for benign samples. Finally, RF was identified as being the most accurate model for intrusion detection, providing good accuracy across the two classes. KNN was also very efficient in this case, especially in the identification of attacks. While LR has been valuable for the purposes of this paper, some problems have been identified with its handling of benign traffic and the overall possibility of more false positives. According to the above findings, RF seems to be the most appropriate for real-time IDSs in an edge-computing-based cyber–physical system.

The confusion matrices of KNN, RF, and LR, as shown in Figures 5–7, present the analysis results of the classification models on the identification of benign and malicious traffic. For KNN, the confusion matrix depicts that the model has successfully classified benign traffic with 79,476 TP and 1608 FP. This leads to an accuracy of 98% and a specificity of 99% for benign instances, which show that KNN is accurate enough in identifying benign traffic. However, for the attack detection, the model has a moderate number of false negatives (606), which means that some of the attacks were not detected at all, which may be an issue in critical applications where all possible threats have to be flagged.

The confusion matrix of RF proves that the model has good accuracy in general. For benign traffic, the true positives are at 52,737, while the false positives are at 1319, resulting in a high level of precision of 0.98 for benign classification through RF. Even more critical, in the case of attack detection, RF has a small number of false negatives equal to 111, so the recall of the algorithm is 1.00. As demonstrated, this makes RF more reliable in detecting attacks with low false negative rates and in accurately categorizing legitimate traffic. From the confusion matrix, the LR model identifies the attacks reasonably well, while four times more false positives are recorded from the benign traffic when compared to both KNN and RF. The model found 47,939 true positives for benign traffic and 6117 false positives; this means it requires more time to analyze, thus giving a lower precision of 0.92. Moreover, it made 2047 false negatives to attacks, and this affected its recall in providing complete details of all possible threats. The larger ratio of false positive cases makes one believe that LR might be classifying benign traffic as malicious more often than it should, which might be disastrous, especially in environments where minimizing false alarms is important.

The comparative analysis of the ROC and precision–recall curves, as shown in Figures 8–10, gives a detailed insight into the results of the KNN, RF, and LR models. In the case of KNN, the optimum results can be observed from the ROC as well as the precision–recall curves, where the obtained values of both curves are equal to one. The ROC curve of true positive and false positive rates gives us an insight that KNN has a sensitivity and specificity of 100% at different thresholds, meaning that KNN does not produce any false positives or false negatives in the classification of benign and malicious traffic. The precision–recall curve also depicts the value of 1.00, which also confirms that KNN has the perfect value of precision and recall. This seems to imply that KNN is very efficient in differentiating between benign and attack instances with no degradation in its capability. In terms of both the ROC and precision–recall curve, RF also showed the best result of one. The results of the ROC suggest that, for RF, there is a perfect distinction between benign traffic and attack traffic and that, even at a high TPR, the FP rate is very low.

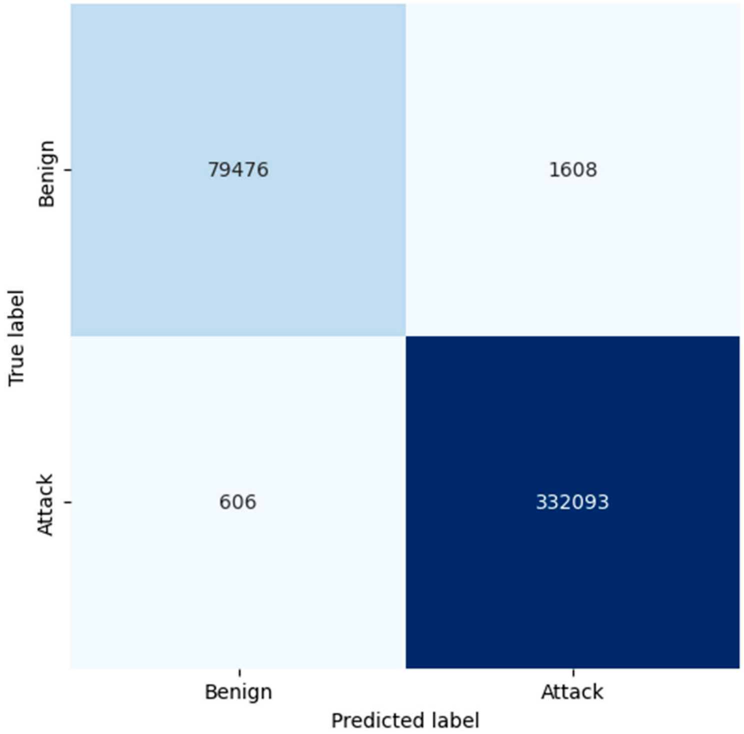

Figure 5. Confusion matrix generated by KNN.

The AUC of the precision–recall curve also stands at 1, which confirms that RF has the potential of achieving both high precision and recall, which means that all instances, whether they are benign or an attack, will be accurately classified. The above evaluation of RF in these measures shows that it is a reliable intrusion detection model for detecting intrusion and assessing the intrusion tolerance of complex systems. Logistic regression, though slightly lower in terms of accuracy compared to the best-performing KNN and RF algorithms, is also quite good. The area under the curve in the case of LR is 0.98, which means that the test is very sensitive and specific but not 100%. The precision–recall curve shows 0.99 and depicts the aspect of precision and recall but a slightly lower score compared to KNN and RF algorithms. This suggests that, even though LR is efficient in identifying the attacks and correctly categorizing the benign traffic, there is a slightly lower performance compared to the other models.

The novelty of our framework lies in its multicriteria integration of network traffic analysis, clustering, machine-learning-based intrusion detection, and edge computing for real-time security enhancement. We integrate k-means and DBSCAN for network traffic clustering, allowing anomaly detection in unlabeled data. We then employ KNN, RF, and LR for supervised intrusion detection, enhancing the ability to classify cyber threats with high accuracy. Instead of relying solely on cloud-based analysis, we provide a concept of task distribution to edge nodes (e.g., routers, fog nodes, and gateways) for real-time security monitoring. The traffic analysis through clustering using k-means and DBSCAN successfully revealed outstanding traffic clusters and anomalies in the network. K-means clustering was seen to be very effective for clustering the dataset into easily distinguished clusters, which helped in the profiling of traffic behavior well. The concept of the silhouette score was used here to justify the acceptable number of clusters, which is 7, with a silhouette score of 0.612, which gave a reasonable measure/degree of cluster separation to identify the traffic patterns with a reasonable level of accuracy. However, DBSCAN clustering, which allowed noise and strange forms of clusters, showed regions of low density that may indicate unusual or malicious activity. Even though the silhouette score was significantly lower with DBSCAN at 0.473, its capabilities of finding outliers and noise would be beneficial to the k-means clustering in offering a robust analysis of the network traffic-related patterns.

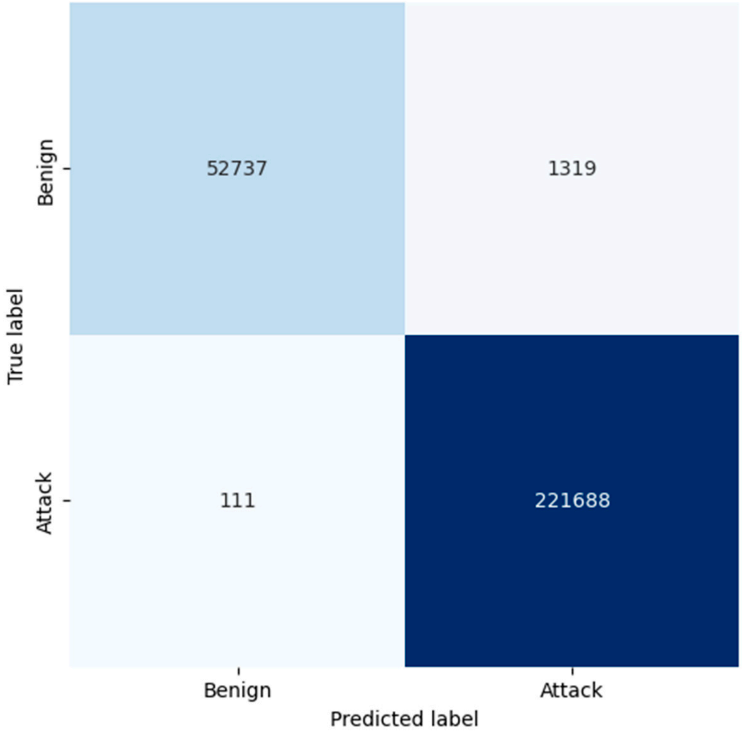

Figure 6. Confusion matrix generated by RF.

The subsequent intrusion detection analysis using KNN, RF, and LR models presented here proved the effectiveness of these approaches for detecting benign and malicious traffic. KNN and RF were almost flawless with all the metrics, ROC and precision–recall, showing how efficient these algorithms are in classifying traffic and identifying potential threats. As it unveiled, KNN presented an adequate figure both in terms of precision and recall; therefore, the use of this model in real-time intrusion detection was suitable. Between the four models tested, RF came out as the most accurate in identifying all the malicious activities due to its ability to accommodate complicated data patterns and its low number of false negatives. Compared to that of LR, the performance of other algorithms, although slightly worse, was still satisfactory, especially in terms of attack detection with high accuracy. Another key area for integration is that edge computing becomes critical for managing and analyzing network traffic data in a non-stop manner and in real time. The system is able to stay adaptive to the high volumes of data produced in industrial IoT settings because edge computing is used to manage the data loads away from central servers.

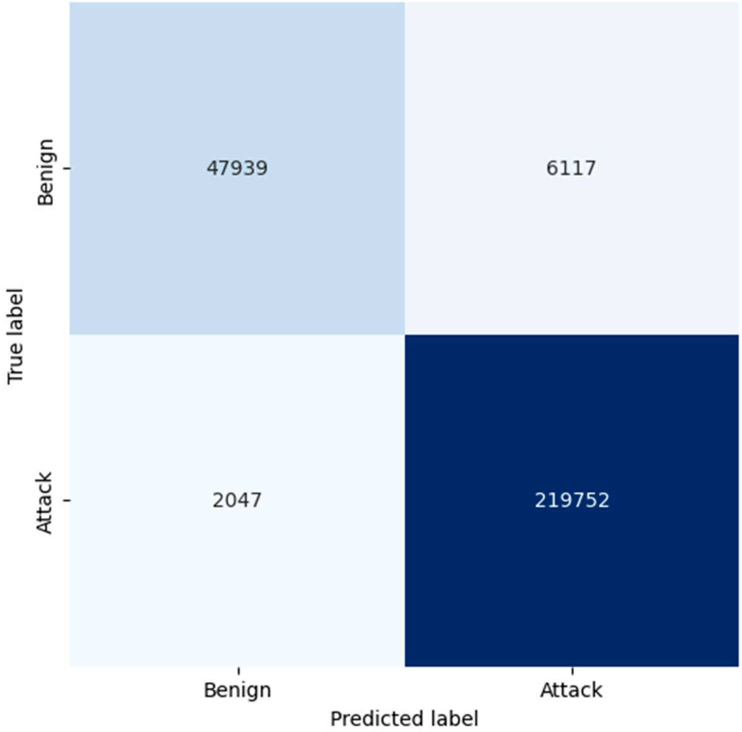

Figure 7. Confusion matrix generated by LR.

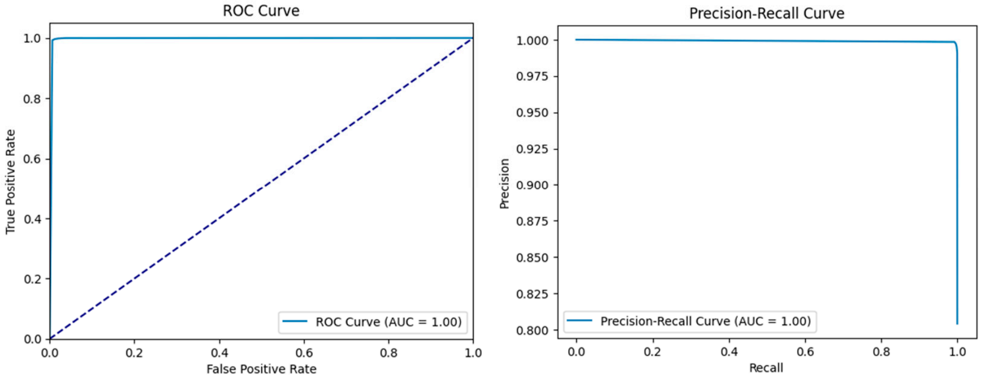

Figure 8. ROC and precision–recall curves for KNN.

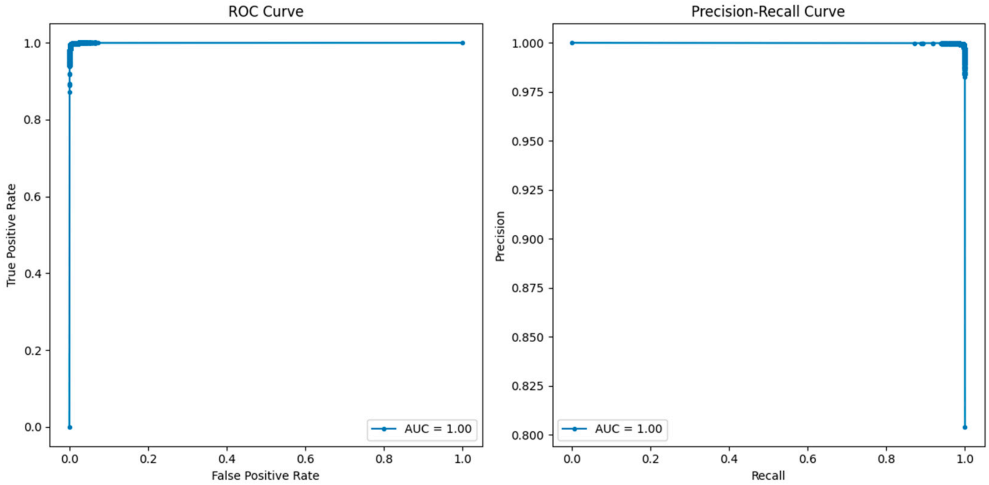

Figure 9. ROC and precision–recall curves for RF.

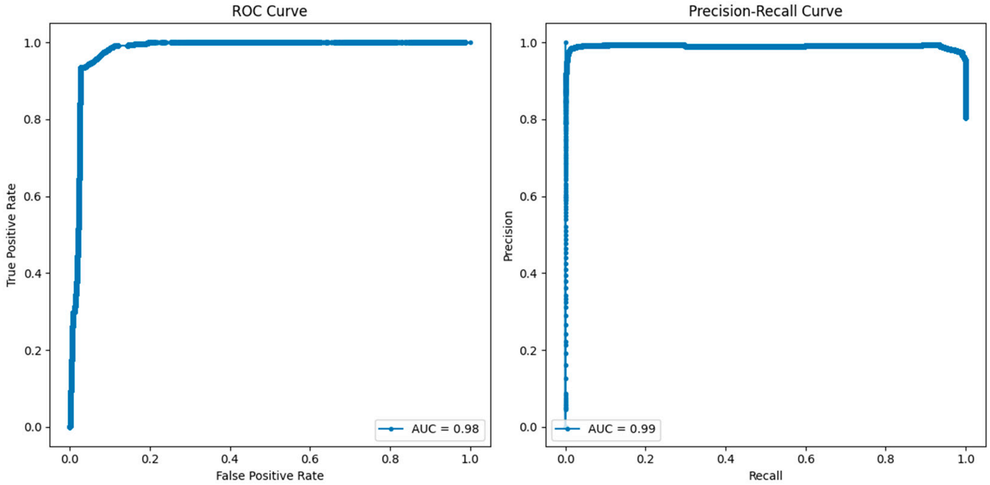

Figure 10. ROC and precision–recall curves for LR.

5. Conclusions

This research work introduces an edge-computing-centered integrated solution for network traffic analysis and intrusion detection that improve CSP security in IIoT. The k-means and DBSCAN algorithms are used in the framework to analyze traffic patterns and classify them as normal; if any traffic pattern deviates from the normal traffic patterns, then the traffic pattern is classified as a possible security threat. The clusters generated by k-means clustering have a silhouette score of 0.612, thus proving that this method can effectively classify data into proper clusters, while the DBSCAN clustering algorithm has a silhouette score of 0.473, thereby showing the ability of the algorithm to detect noise and outliers in network traffic. These clustering methods are thus used as a starting point for subsequent security analysis to improve traffic behavior understanding. In intrusion detection, the model was used, and the application of machine-learning models was carried out and closely tested. KNN and RF models were also at the top of the ranking, receiving 1.00 for both the ROC and precision–recall scores. Just slightly less accurate was logistic regression, which still had an ROC of 0.98 and a precision–recall score of 0.99. These results are a clear indication of the viability of the machine-learning technique in detecting intrusions with few false positives and false negatives, and, thus, increase the reliability of the detector. The confusion matrices support this and indicate, for example, that RF specifically was significantly better at correctly categorizing benign and malicious traffic. The use of edge computing integration is an important element of the framework. By carrying out most of the processing on the edge, the system reduces the latency, hence making it possible to provide real-time decisions and freeing the cloud center resources. This architecture is appropriate for use in industrial IoT scenarios since a high response time to threats is required to sustain operation. Edge computing also brings scalability since the system can easily expand as the IoT networks grow without much negative impact on performance, let alone security issues.

In future, we will focus on refining ML models by integrating time-series analysis and advanced communication protocols. We aim to explore hybrid algorithms to improve risk assessment and anomaly identification, especially against more sophisticated attack types in CPSs. We also plan to implement fog-computing-based fuzzy logic systems to optimize 5G communication technology performance in IIoT.

Author Contributions: Conceptualization, T.Z. and Z.A.; data curation, Z.A. and N.K.; formal analysis, T.Z., Z.A., A.A. (Aigul Adamova), N.K. and A.A. (Assel Abdildayeva); funding acquisition, T.Z.; investigation, Z.A., A.A. (Aigul Adamova), N.K. and A.A. (Assel Abdildayeva); methodology, A.A. (Aigul Adamova) and N.K.; project administration, T.Z. and A.A. (Assel Abdildayeva); resources, T.Z., A.A. (Aigul Adamova) and A.A. (Assel Abdildayeva); software, Z.A. and N.K.; supervision, T.Z. and A.A. (Aigul Adamova); validation, Z.A., N.K. and A.A. (Assel Abdildayeva); visualization, Z.A., A.A. (Aigul Adamova), N.K. and A.A. (Assel Abdildayeva); writing—original draft, T.Z., Z.A. and N.K.; writing—review and editing, A.A. (Aigul Adamova) and A.A. (Assel Abdildayeva). All authors have read and agreed to the published version of the manuscript.

Funding: This research has been funded by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan (grant No. AP23489127).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset supporting the findings of this study is publicly available on a Kaggle website with the title NF-ToN-IoT-V2 [48].

Acknowledgments: The authors sincerely appreciate the support provided by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan for funding this research under Grant No. AP23489127.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Fernández-Caramés, T.M.; Fraga-Lamas, P. Use Case Based Blended Teaching of IIoT Cybersecurity in the Industry 4.0 Era. *Appl. Sci.* **2020**, *10*, 5607. [CrossRef]
2. Zhu, E.; Wang, H.; Zhang, Y.; Zhang, K.; Liu, C. PHEE: Identifying influential nodes in social networks with a phased evaluation-enhanced search. *Neurocomputing* **2024**, *572*, 127195. [CrossRef]
3. Peng, K.; Zhao, B.; Bilal, M.; Xu, X.; Nayyar, A. QoS-Aware Cloud-Edge Collaborative Micro-Service Scheduling in the IIoT. *Hum.-Centric Comput. Inf. Sci.* **2023**, *13*. [CrossRef]
4. Zhang, G.; Wei, X.; Tan, X.; Han, Z.; Zhang, G. AoI Minimization Based on Deep Reinforcement Learning and Matching Game for IoT Information Collection in SAGIN. *IEEE Trans. Commun.* **2025**. *early access*. [CrossRef]
5. Qiao, Y.; Lü, J.; Wang, T.; Liu, K.; Zhang, B.; Snoussi, H. A Multihead Attention Self-Supervised Representation Model for Industrial Sensors Anomaly Detection. *IEEE Trans. Ind. Inform.* **2024**, *20*, 2190–2199. [CrossRef]
6. Peter, O.; Pradhan, A.; Mbohwa, C. Industrial internet of things (IIoT): Opportunities, challenges, and requirements in manufacturing businesses in emerging economies. *Procedia Comput. Sci.* **2023**, *217*, 856–865. [CrossRef]
7. Alotaibi, B. A Survey on Industrial Internet of Things Security: Requirements, Attacks, AI-Based Solutions, and Edge Computing Opportunities. *Sensors* **2023**, *23*, 7470. [CrossRef]
8. Diène, B.; Rodrigues, J.J.P.C.; Diallo, O.; Ndoye, E.H.M.; Korotaev, V.V. Data management techniques for Internet of Things. *Mech. Syst. Signal Process.* **2020**, *138*, 106564. [CrossRef]
9. Javed, A.; Robert, J.; Heljanko, K.; Främling, K. IoTEF: A Federated Edge-Cloud Architecture for Fault-Tolerant IoT Applications. *J. Grid Comput.* **2020**, *18*, 57–80. [CrossRef]
10. Hajda, J.; Jakuszewski, R.; Ogonowski, S. Security Challenges in Industry 4.0 PLC Systems. *Appl. Sci.* **2021**, *11*, 9785. [CrossRef]
11. Liu, Y.; Li, W.; Dong, X.; Ren, Z. Resilient Formation Tracking for Networked Swarm Systems Under Malicious Data Deception Attacks. *Int. J. Robust Nonlinear Control* **2024**, *35*, 2043–2052. [CrossRef]
12. Li, C.; He, A.; Liu, G.; Wen, Y.; Chronopoulos, A.T.; Giannakos, A. RFL-APIA: A Comprehensive Framework for Mitigating Poisoning Attacks and Promoting Model Aggregation in IIoT Federated Learning. *IEEE Trans. Ind. Inform.* **2024**, *20*, 12935–12944. [CrossRef]
13. Abosuliman, S.S. Deep learning techniques for securing cyber-physical systems in supply chain 4.0. *Comput. Electr. Eng.* **2023**, *107*, 108637. [CrossRef]
14. Wang, E.; Yang, Y.; Wu, J.; Liu, W.; Wang, X. An Efficient Prediction-Based User Recruitment for Mobile Crowdsensing. *IEEE Trans. Mob. Comput.* **2018**, *17*, 16–28. [CrossRef]
15. Jiang, H.; Ji, P.; Zhang, T.; Cao, H.; Liu, D. Two-Factor Authentication for Keyless Entry System via Finger-Induced Vibrations. *IEEE Trans. Mob. Comput.* **2024**, *23*, 9708–9720. [CrossRef]

16. Alguliyev, R.; Imamverdiyev, Y.; Sukhostat, L. Cyber-physical systems and their security issues. *Comput. Ind.* **2018**, *100*, 212–223. [CrossRef]
17. Al-Quayed, F.; Ahmad, Z.; Humayun, M. A Situation Based Predictive Approach for Cybersecurity Intrusion Detection and Prevention Using Machine Learning and Deep Learning Algorithms in Wireless Sensor Networks of Industry 4.0. *IEEE Access* **2024**, *12*, 34800–34819. [CrossRef]
18. Zhou, W.; Xia, C.; Wang, T.; Liang, X.; Lin, W.; Li, X.; Zhang, S. HIDIM: A novel framework of network intrusion detection for hierarchical dependency and class imbalance. *Comput. Secur.* **2025**, *148*, 104155. [CrossRef]
19. Li, T.; Kouyoumdjieva, S.T.; Karlsson, G.; Hui, P. Data collection and node counting by opportunistic communication. In Proceedings of the 2019 IFIP Networking Conference (IFIP Networking), Warsaw, Poland, 20–22 May 2019; pp. 1–9.
20. Goldstein, A.; Johanndeiter, T.; Frank, U. *Business Process Runtime Models: Towards Bridging the Gap Between Design, Enactment, and Evaluation of Business Processes*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 17, ISBN 1025701803742.
21. Nankya, M.; Chataut, R.; Akl, R. Securing Industrial Control Systems: Components, Cyber Threats, and Machine Learning-Driven Defense Strategies. *Sensors* **2023**, *23*, 8840. [CrossRef]
22. Umran, S.M.; Lu, S.; Abduljabbar, Z.A.; Zhu, J.; Wu, J. Secure Data of Industrial Internet of Things in a Cement Factory Based on a Blockchain Technology. *Appl. Sci.* **2021**, *11*, 6376. [CrossRef]
23. Zhang, X.; Hou, D.; Xiong, Z.; Liu, Y.; Wang, S.; Li, Y. EALLR: Energy-Aware Low-Latency Routing Data Driven Model in Mobile Edge Computing. *IEEE Trans. Consum. Electron.* **2024**. *early access*. [CrossRef]
24. Li, X.; Liu, Y.; Ji, H.; Zhang, H.; Leung, V.C.M. Optimizing resources allocation for fog computing-based internet of things networks. *IEEE Access* **2019**, *7*, 64907–64922. [CrossRef]
25. Sun, G.; Liao, D.; Zhao, D.; Xu, Z.; Yu, H. Live Migration for Multiple Correlated Virtual Machines in Cloud-Based Data Centers. *IEEE Trans. Serv. Comput.* **2018**, *11*, 279–291. [CrossRef]
26. Algarni, A.; Acarer, T.; Ahmad, Z. An Edge Computing-Based Preventive Framework With Machine Learning- Integration for Anomaly Detection and Risk Management in Maritime Wireless Communications. *IEEE Access* **2024**, *12*, 53646–53663. [CrossRef]
27. Cao, K.; Hu, S.; Shi, Y.; Colombo, A.; Karnouskos, S.; Li, X. A Survey on Edge and Edge-Cloud Computing Assisted Cyber-Physical Systems. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7806–7819. [CrossRef]
28. Sun, G.; Wang, Z.; Su, H.; Yu, H.; Lei, B.; Guizani, M. Profit Maximization of Independent Task Offloading in MEC-Enabled 5G Internet of Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 16449–16461. [CrossRef]
29. Safa, N.S.; Maple, C.; Furnell, S.; Azad, M.A.; Perera, C.; Dabbagh, M.; Sookhak, M. Deterrence and prevention-based model to mitigate information security insider threats in organisations. *Futur. Gener. Comput. Syst.* **2019**, *97*, 587–597. [CrossRef]
30. Babar, M.; Khan, M.S.; Habib, U.; Shah, B.; Ali, F.; Song, D. Scalable Edge Computing for IoT and Multimedia Applications Using Machine Learning. *Hum.-Centric Comput. Inf. Sci.* **2021**, *11*, 41. [CrossRef]
31. Ma, Y.; Ma, R.; Lin, Z.; Zhang, R.; Cai, Y.; Wu, W.; Wang, J. Improving Age of Information for Covert Communication With Time-Modulated Arrays. *IEEE Internet Things J.* **2025**, *12*, 1718–1731. [CrossRef]
32. Yao, C.; Yang, Y.; Yin, K.; Yang, J. Traffic Anomaly Detection in Wireless Sensor Networks Based on Principal Component Analysis and Deep Convolution Neural Network. *IEEE Access* **2022**, *10*, 103136–103149. [CrossRef]
33. Lin, W.; Xia, C.; Wang, T.; Zhao, Y.; Xi, L.; Zhang, S. Input and Output Matter: Malicious Traffic Detection with Explainability. *IEEE Netw.* **2024**, *39*, 259–267. [CrossRef]
34. Shah, K.; Patel, H.; Sanghvi, D.; Shah, M. A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification. *Augment. Hum. Res.* **2020**, *5*, 12. [CrossRef]
35. Peppes, N.; Daskalakis, E.; Alexakis, T.; Adamopoulou, E.; Demestichas, K. Performance of Machine Learning-Based Multi-Model Voting Ensemble Methods for Network Threat Detection in Agriculture 4.0. *Sensors* **2021**, *21*, 7475. [CrossRef]
36. Venkatesan, V.K.; Ramakrishna, M.T.; Izonin, I.; Tkachenko, R.; Havryliuk, M. Efficient Data Preprocessing with Ensemble Machine Learning Technique for the Early Detection of Chronic Kidney Disease. *Appl. Sci.* **2023**, *13*, 2885. [CrossRef]
37. Farnaaz, N.; Jabbar, M.A. Random Forest Modeling for Network Intrusion Detection System. *Procedia Comput. Sci.* **2016**, *89*, 213–217. [CrossRef]
38. Alrashdi, I.; Alqazzaz, A.; Aloufi, E.; Alharthi, R.; Zohdy, M.; Ming, H. AD-IoT: Anomaly detection of IoT cyberattacks in smart city using machine learning. In Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference, CCWC 2019, Las Vegas, NV, USA, 7–9 January 2019; pp. 305–310.
39. Mozaffaree Pour, N.; Oja, T. Prediction Power of Logistic Regression (LR) and Multi-Layer Perceptron (MLP) Models in Exploring Driving Forces of Urban Expansion to Be Sustainable in Estonia. *Sustainability* **2021**, *14*, 160. [CrossRef]
40. Muthuramalingam, S.; Bharathi, A.; Rakesh Kumar, S.; Gayathri, N.; Sathiyaraj, R.; Balamurugan, B. Iot based intelligent transportation system (iot-its) for global perspective: A case study. In *Internet of Things and Big Data Analytics for Smart Generation*; Springer: Cham, Switzerland, 2019; pp. 279–300.
41. Kumar, A. Cybersecurity Threat Detection using Machine Learning and Network Analysis. *J. Artif. Intell. Gen. Sci.* **2024**, *1*, 124–131. [CrossRef]

42. Sangaiah, A.K.; Javadpour, A.; Pinto, P. Towards data security assessments using an IDS security model for cyber-physical smart cities. *Inf. Sci.* **2023**, *648*, 119530. [CrossRef]
43. Ni, C.; Li, S.C. Machine learning enabled Industrial IoT Security: Challenges, Trends and Solutions. *J. Ind. Inf. Integr.* **2024**, *38*, 100549. [CrossRef]
44. Wisdom, D.D.; Vincent, O.R.; Igulu, K.; Hyacinth, E.A.; Christian, A.U.; Oduntan, O.E.; Hauni, A.G. Industrial IoT Security Infrastructures and Threats. In *Communication Technologies and Security Challenges in IoT: Present and Future*; Springer: Singapore, 2024; pp. 369–402.
45. Yafooz, W.M.S.; Bakar, Z.B.A.; Fahad, S.K.A. Business Intelligence Through Big Data Analytics, Data Mining and Machine Learning. In *Data Management, Analytics and Innovation, Proceedings of ICDMAI 2019, Kuala Lumpur, Malaysia, 18–20 January 2019*; Springer: Singapore, 2020; Volume 1016, pp. 217–230.
46. Kumari, A.; Patel, R.K.; Sukharamwala, U.C.; Tanwar, S.; Raboaca, M.S.; Saad, A.; Tolba, A. AI-Empowered Attack Detection and Prevention Scheme for Smart Grid System. *Mathematics* **2022**, *10*, 2852. [CrossRef]
47. Balaji, B.S.; Paja, W.; Antonijevic, M.; Stoean, C.; Bacanin, N.; Zivkovic, M. IoT Integrated Edge Platform for Secure Industrial Application with Deep Learning. *Hum.-Centric Comput. Inf. Sci.* **2023**, *13*. [CrossRef]
48. Kaggle. NF-ToN-IoT-V2. Available online: https://www.kaggle.com/datasets/dhoogla/nftoniotv2/data (accessed on 16 August 2024).
49. Singh, V.; Gupta, I.; Jana, P.K. A novel cost-efficient approach for deadline-constrained workflow scheduling by dynamic provisioning of resources. *Future Gener. Comput. Syst.* **2018**, *79*, 95–110. [CrossRef]
50. Matni, N.; Moraes, J.; Oliveira, H.; Rosário, D.; Cerqueira, E. Lorawan gateway placement model for dynamic internet of things scenarios. *Sensors* **2020**, *20*, 4336. [CrossRef]
51. Deng, D. DBSCAN Clustering Algorithm Based on Density. In Proceedings of the 2020 7th International Forum on Electrical Engineering and Automation (IFEEA), Hefei, China, 25–27 September 2020; pp. 949–953.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.