Ensemble-Based Deep Learning Models for Enhancing IoT Intrusion Detection

Ammar Odeh * and Anas Abu Taleb

Department of Computer Science, Princess Sumaya University of Technology, Amman 1196, Jordan; a.abutaleb@psut.edu.jo
* Correspondence: a.odeh@psut.edu.jo

Abstract: Cybersecurity finds widespread applications across diverse domains, encompassing intelligent industrial systems, residential environments, personal gadgets, and automobiles. This has spurred groundbreaking advancements while concurrently posing persistent challenges in addressing security concerns tied to IoT devices. IoT intrusion detection involves using sophisticated techniques, including deep learning models such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and anomaly detection algorithms, to identify unauthorized or malicious activities within IoT ecosystems. These systems continuously monitor and analyze network traffic and device behavior, seeking patterns that deviate from established norms. When anomalies are detected, security measures are triggered to thwart potential threats. IoT intrusion detection is vital for safeguarding data integrity, ensuring users’ privacy, and maintaining critical systems’ reliability and safety. As the IoT landscape evolves, effective intrusion detection mechanisms become increasingly essential to mitigate the ever-growing spectrum of cyber threats. Practical security approaches, notably deep learning-based intrusion detection, have been introduced to tackle these issues. This study utilizes deep learning models, including convolutional neural networks (CNNs), long short-term memory (LSTM), and gated recurrent units (GRUs), while introducing an ensemble deep learning architectural framework that integrates a voting policy within the model’s structure, thereby facilitating the computation and learning of hierarchical patterns. In our analysis, we compared the performance of ensemble deep learning classifiers with traditional deep learning techniques. The standout models were CNN-LSTM and CNN-GRU, achieving impressive accuracies of 99.7% and 99.6%, along with exceptional F1-scores of 0.998 and 0.997, respectively.

Keywords: intrusion detection system; Internet of Things; cybersecurity; cyber-physical systems; cyberattacks; cybercrime; intrusion detection

1. Introduction

The Internet of Things (IoT) represents a transformative concept where everyday objects, devices, and appliances are interconnected, enabling them to collect and exchange data [1]. This vast network extends the Internet’s reach beyond traditional devices like computers and smartphones, encompassing everything from household appliances and wearables to vehicles and industrial machinery. This seamless connectivity offers unparalleled convenience, fostering smarter cities, enhanced healthcare, and more efficient industries—Figure 1 shows the top IoT applications [2].

However, with this rapid expansion and integration of devices into daily life, several challenges arise, particularly in security [3,4]. The vast number of connected devices presents a large attack surface, making them potential entry points for malicious actors. Moreover, the lack of standardization, default insecure settings, and limited computational power in many IoT devices compound these security vulnerabilities. As a result, there is an imperative need for robust security solutions to safeguard the ever-evolving IoT landscape [5].
there is an imperative need for robust security solutions to safeguard the ever-evolving IoT landscape [5].

Furthermore, the IoT ecosystem’s heterogeneous nature, characterized by various manufacturers, protocols, and software stacks, complicates establishing a unified security approach. Many IoT devices, designed primarily for functionality and cost-effectiveness, often lack regular software updates, exposing them to known security threats for prolonged periods [6,7]. Data privacy is another pressing concern. As these devices continually collect vast amounts of data, sensitive information can be accessed or misused, threatening individual privacy and corporate confidentiality [8].

Addressing these security challenges is paramount to ensure that the IoT realizes its full potential without compromising user trust and safety [9]. As the adoption of IoT continues to surge, there is an increasing emphasis on developing sophisticated security measures, including advanced intrusion detection systems and adaptive threat response mechanisms. Only by prioritizing and innovating in the realm of security can the promise of a truly connected, innovative, and safe world be achieved [10,11].

Moreover, the physical nature of IoT devices adds another layer of vulnerability. Unlike purely digital platforms, these devices can be physically tampered with, leading to breaches not just in data but also in their operational integrity. Consider smart infrastructure in cities or hospital health devices; tampering could have real-world, life-threatening consequences [12].

This interconnectedness, while being the strength of the IoT, also becomes its Achilles’ heel. A breach in one device can potentially cascade, affecting a network of interconnected systems, emphasizing the need for holistic security frameworks. Collaboration across industries, manufacturers, and regulatory bodies is essential to develop and enforce standards that ensure the resilience and safety of the IoT ecosystem. As research and development forge ahead, integrating security from the inception of device design, rather than as an afterthought, will be crucial in defining the future of a secure and efficient IoT landscape [13–15]. The author in [16,17] introduces an enhanced aggregate segmentation mask RCNN model (AS Mask RCNN) for grading mixed aggregates. The study conducted three different experiments and found that the AS Mask RCNN achieved an impressive accuracy of over 89.13% across all experimental scenarios. Compared to the faster RCNN
and mask R-CNN models, it demonstrated an accuracy improvement of 8.85%. It reduced the processing time for single image segmentation by 1.29 s, making it suitable for near real-time field detection requirements. The paper also presents a self-developed noncontact testing platform for aggregate grading that can be applied in complex environments. This platform facilitates digital, automated, and intelligent noncontact grading of mixed aggregates, ultimately enhancing the accuracy of aggregate grading testing and supporting the high-quality development of reservoir dam construction in China. The author’s work focuses on the significant role of the human microbiome in predicting certain diseases. They acknowledge the challenges posed by limited samples and high-dimensional features in microbiome data for machine learning methods. The author introduces a novel ensemble deep-learning disease prediction method to address this. The approach combines unsupervised and supervised learning techniques. It starts with unsupervised deep learning to discover potential sample representations. Then, these representations are used to develop a disease-scoring strategy, creating informative features for ensemble analysis. A score selection mechanism is implemented to ensure optimal ensemble performance, and performance-enhancing features are incorporated with the original data [18].

Certainly, summarizing the main contributions into three points:

- we introduce a flexible and highly efficient approach that utilizes ensemble-based deep learning models to swiftly and accurately detect intrusions in IoT environments while mitigating false positives and negatives;
- we evaluate and characterize the performance of deep learning algorithms;
- we provide a systematic and comparative experimental analysis.

1.1. IoT Security Challenges

The IoT (Internet of Things) brings about a revolution in connectivity, enabling devices to communicate seamlessly. However, with this increased connectivity comes a myriad of security challenges. Figure 2 depicts some of the most pressing security challenges associated with the IoT.

**Figure 2.** IoT security challenges.

1.1.1. Lack of Physical Security

The absence of robust physical safeguards on IoT devices makes them vulnerable to unauthorized access. Devices stationed in isolated locations over extended periods are particularly susceptible to tampering. The ease with which attackers can exploit IoT devices with minimal physical protection poses significant security challenges [19].

Consider, for instance, the potential for IoT devices to be compromised via malware-laden USB flash drives. While it is incumbent upon IoT device manufacturers to prioritize
their products’ physical security, engineering secure yet cost-effective transmitters and sensors remains a daunting task for them [20].

1.1.2. Lack of Standardization

A diverse range of manufacturers produces IoT devices, each adhering to unique standards and protocols. This absence of standardized security measures can lead to vulnerabilities, offering potential entry points for exploitation. Furthermore, this fragmentation in manufacturing practices and protocols complicates establishing a cohesive security framework for the IoT. Since devices might communicate differently and prioritize varied security aspects, ensuring compatibility and security across the board becomes challenging. This disjointed landscape hinders interoperability and makes it harder to deploy universal security patches or updates. For users, this means a heightened risk, as one weak device can compromise the security of an entire connected network. As the IoT ecosystem continues to expand, industry-wide collaboration is urgently needed to establish and enforce consistent security standards, ensuring a safer and more integrated digital future [21,22].

1.1.3. Lack of Visibility

For I.T. teams, obtaining a comprehensive view of all devices on the network is daunting, primarily because numerous devices are not cataloged in the I.T. inventory. Often overlooked by I.T. teams, devices such as coffee machines, ventilation systems, and air conditioners are not typically tracked [23].

If security teams are unaware of the devices connected to the network, they cannot effectively prevent breaches. The insufficient visibility into IoT devices complicates the I.T. department’s task of accurately identifying and monitoring assets that require protection [24].

1.1.4. Data Privacy and Integrity

In IoT, data privacy emerges as a paramount security concern. User data traverses many devices, from medical equipment divulging patient details to intelligent toys and wearables revealing personal information. To illustrate this, a cybercriminal could potentially harvest corporate information, exposing, selling, or leveraging it to blackmail the proprietor [25].

1.1.5. Physical Security Threats

Given their physical nature, IoT devices are inherently vulnerable to direct interference and manipulation. Malicious actors can exploit these devices by gaining hands-on access to their hardware components, potentially altering their functionalities or extracting sensitive data. This tangible aspect of IoT emphasizes the importance of digital and physical security measures to protect against unauthorized interventions [26].

1.1.6. Insecure Data Storage and Transmission

A significant number of IoT devices lack data encryption for both stored and transmitted information. This oversight exposes the data, allowing potential eavesdroppers to intercept and access it without authorization. Such lax security measures underscore the pressing need for enhanced encryption protocols in the IoT landscape to safeguard against breaches and unauthorized intrusions [27]. Additionally, the absence of robust encryption practices exacerbates the risk of man-in-the-middle attacks, where malicious actors can intercept and potentially alter data as it is being transmitted between devices. This not only compromises the confidentiality of the information but also its integrity. Furthermore, with the growing reliance on IoT devices in critical sectors such as healthcare, transportation, and energy, the consequences of unauthorized data access could be dire, ranging from personal privacy breaches to large-scale infrastructure disruptions. For these
reasons, manufacturers must prioritize and implement advanced encryption techniques, ensuring both the security and trustworthiness of IoT device communications [28].

1.1.7. Botnet Attacks

A significant security issue with IoT pertains directly to the devices themselves. Their inherent security vulnerabilities make them prime targets for botnet infiltrations.

Essentially, a botnet is an ensemble of machines compromised by malware. Attackers harness these compromised machines to flood targets with overwhelming request traffic. Unlike conventional computers, IoT devices often lack regular security updates, heightening their susceptibility to malware exploits. Consequently, malicious actors can swiftly transform these devices into botnets, becoming conduits for vast request traffic [29,30].

1.1.8. Ransomware

In the context of IoT security, ransomware poses a significant threat by encrypting and barring access to vital files. To regain access, hackers typically demand a ransom in exchange for the decryption key [31].

While currently uncommon, IoT devices with subpar security might become future victims of ransomware. As the value and dependence on healthcare devices, smart homes, and other intelligent appliances grow, they could become increasingly attractive targets, especially given their critical importance to users [32].

1.2. Intrusion Detection Systems (IDS)

Intrusion detection refers to identifying malicious activities carried out against information systems. These malevolent acts, termed intrusions, are efforts to gain unauthorized access to a computer system. Intruders can be categorized into two main types: internal and external. Internal intruders are individuals within the network who, despite having some legitimate access, aim to elevate their access privileges to misuse resources they are not authorized for. In contrast, external intruders are individuals outside the network aiming to infiltrate it and access system information without permission [33].

Both types of intruders pose distinct challenges. Internal intruders, already having some degree of legitimate access, can exploit vulnerabilities from within, making their actions more complicated to detect. Their familiarity with the system’s architecture and potential weak points can make their intrusions more targeted and potentially more damaging. On the other hand, external intruders, although initially lacking access, often employ a wide range of techniques, from brute-force attacks to sophisticated phishing schemes, to breach the system’s defenses [34].

Moreover, the rise of IoT devices and the expanding digital landscape have further complicated intrusion detection. With more entry points and a diverse range of devices, networks are more susceptible than ever. This underscores the importance of robust security measures, continuous system monitoring, and regular updates to defend against evolving threats. Additionally, organizations must foster a culture of security awareness, ensuring that every internal or external user is well-informed about potential risks and best practices to mitigate them.

The evolving dynamics of cyber threats necessitate an adaptive and layered approach to security. Intrusion detection systems (IDS) are just one component of a comprehensive cybersecurity strategy. Beyond simple detection, the focus has shifted towards intrusion prevention systems (IPS) that not only detect but also take proactive measures to prevent unauthorized access [35,36].

Furthermore, with the integration of artificial intelligence and machine learning in security systems, there is an opportunity to predict and identify novel threats before they manifest. These predictive systems analyze patterns and behaviors, allowing them to flag anomalous activities even if they do not match known threat signatures.

Yet, technology alone is not the panacea. Human factors play a significant role in security breaches. Regular training sessions, workshops, and awareness campaigns should
be organized for employees and users. This ensures that they are aware of the potential risks and equipped with the knowledge to recognize and report suspicious activities [37].

The principle of least privilege (PoLP) should be strictly adhered to, meaning that users should only be granted access to the information and resources necessary for their specific tasks, reducing the potential damage of an internal intrusion.

In a world where cyber threats continually evolve, staying a step ahead is crucial. This requires cutting-edge technology, strategic planning, and an informed and vigilant user base. By integrating these elements, organizations can fortify their defenses, ensuring data integrity and maintaining the trust of their users [38].

2. Intrusion Detection in the Internet of Things

This part examines various literature sources on IDS solutions tailored for the IoT. The intrusion detection systems (IDSs) taxonomy for the Internet of Things (IoT) offers a structured framework to categorize and understand various IDS solutions tailored to IoT environments. Table 1 shows the different categories of IDS categories and subcategories. Here is the breakdown.

<table>
<thead>
<tr>
<th>Table 1. Intrusion detection in the Internet of Things.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IDS Proposals for IoT</strong></td>
</tr>
<tr>
<td>________________________</td>
</tr>
<tr>
<td>Placement strategy</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Detection method</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Security threat</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Validation strategy</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
2.1. Placement Strategy

The placement strategy of an intrusion detection system (IDS) is crucial, as it deter-
mines where in the architecture the IDS operates, influencing its effectiveness, coverage,
and operational cost. For the Internet of Things (IoT) environments, the complexity and
diversity of devices and their specific requirements offer unique challenges [39]. The
placement strategy determines where the IDS is deployed within the IoT infrastructure.

- Edge-based IDS: deployed on edge devices or gateways.
- Cloud-based IDS: utilizes cloud resources for intrusion detection.
- Hybrid: combines both edge and cloud-based approaches.

2.2. Detection Method

This refers to how the IDS detects potential threats [40].

- Signature-based: uses predefined patterns or signatures of known threats.
- Anomaly-based: establishes a “normal” behavior baseline and detects deviations from
  this baseline.
- Specification-based: defines a set of rules or specifications that determine valid behavior.
- Hybrid: combines multiple methods for more comprehensive detection.

2.3. Security Threat

This pertains to the specific types of threats the IDS is designed to detect [41].

- Physical attacks: direct tampering with IoT devices.
- Network attacks: DDoS attacks, man-in-the-middle attacks, etc.
- Software attacks: malware, ransomware, etc.
- Side-channel attacks: exploits that target the physical implementation of IoT systems.

2.4. Validation Strategy

How the effectiveness of the IDS is tested and validated [42].

- Simulation: using software to emulate an IoT environment and test the IDS.
- Testbed: a controlled physical environment where real IoT devices are used.
- Real-world deployment: implementing the IDS in a live IoT environment.
- Theoretical analysis: using mathematical or conceptual models to validate the IDS’s
  effectiveness.

The IDSs for IoT provide a structured way to analyze and compare different IDS
solutions. By understanding where an IDS is placed, how it detects threats, the specific
threats it targets, and how its effectiveness is validated, stakeholders can make informed
decisions about implementing the most appropriate IDS for their specific IoT environment.
Table 1 summerize the Intrusion detection in the Internet of Things.

3. Related Works

Intrusion detection has gained significant prominence in the cybersecurity sector [43].
In recent years, there has been a growing emphasis on employing deep learning (DL)
solutions in this field. Numerous instances of this trend have emerged across various I.T.
domains, including cloud computing [44,45] and computer networking [46,47]. With the
pervasive integration of IoT devices into our daily lives, a substantial portion of recent
research in intrusion detection systems (IDS) has been dedicated to DL solutions within the
IoT domain.

For example, Latif et al. [48] introduced an innovative, lightweight approach centered
on a dense random neural network (DnRaNN) for detecting intrusions in IoT networks.
Their method was rigorously evaluated against the ToN_IoT dataset, yielding outstanding
results in binary and multi-class classification scenarios. Kumar et al. [49] employed the
same dataset to investigate their DL-driven cyber threat modeling framework designed
to automate the detection and extraction of cyber threats in IoT-enabled maritime trans-
Portation systems (MTS). Similar to previous research, they achieved promising binary and multi-class classification results.

In another avenue of intrusion detection, an approach based on an adaptive particle swarm optimization convolutional neural network (APSO-CNN) was proposed [50]. This approach leverages the APSO algorithm to fine-tune the hyperparameters of a one-dimensional CNN autonomously. Evaluation of the N-BaIoT dataset [51] and comparison with three other models demonstrated that this solution consistently outperformed its counterparts across all metrics.

Additionally, initially designed for computer vision tasks such as facial or shape recognition, CNN-based solutions have gained traction and proven effective for network intrusion detection systems (NIDS) [52–54]. Shallow deep learning (DL) methods have also been advanced for IoT IDS. For instance, one approach based on a shallow artificial neural network (ANN) was presented by [55], focusing on the UNSW-15 dataset [56]. Another similar approach termed a multi-layer perceptron (MLP), was proposed by [57] to detect denial of service (DoS) attacks in IoT environments, evaluated on their custom testbed.

Further innovation includes a hybrid approach combining shallow and deep ANNs [58]. Recurrent neural networks (RNNs), particularly long short-term memory (LSTM), have gained popularity. LSTMs are ANNs that preserve hidden states, allowing the model to retain information from previous layers, making them suitable for sequences where each data point depends on its predecessors. RNNs, including LSTMs, have been extensively applied in various domains, such as speech recognition and text generation, and their use is also prevalent in IoT intrusion detection.

For instance, Azumah et al. [59] introduced an LSTM-based approach for intrusion detection in smart home networks, achieving exceptionally high accuracy. In [60], LSTM was employed to detect attacks in Fog computing, using the ISCX2012 dataset [61] and another dataset derived from traffic in an 802.11 network. While the results appeared promising, the authors limited their comparison to another approach based on logistic regression (L.R.), and a more comprehensive evaluation against advanced DL solutions could have provided a more thorough assessment of their work. Additionally, LSTM models have been tested in the automotive environment, as seen in [62], where a manually generated dataset based on traffic from a controller area network (CAN) was utilized as a testbed. The results were excellent and were validated using an open-source CAN dataset.

Ajaeiya et al. [63] presented an intrusion detection system in SDN using random forest. They employed network flow records and their statistics as features for training the machine learning model. To validate its performance, the system was tested with various network attacks, including brute force and reconnaissance attacks.

Hadem et al. [64] combined SVM and selective logging with IP traceback for detecting intrusions in SDN. They reported that their IDS implementation was resource-efficient, leading to savings in memory usage. To evaluate their approach, they conducted experiments using the NSL-KDD dataset, achieving a detection accuracy of 87.74%.

Ye et al. [65] introduced an SVM-based machine learning method for detecting distributed denial of service (DDoS) attacks within SDN. They employed network tuple characteristics as features to identify network protocol attacks, including TCP, UDP, and ICMP attacks. They generated their dataset using network traffic in a controlled testing environment, utilizing tools like hping3. Their findings indicate a high detection accuracy of 95.34% for recognizing UDP flooding attacks.

ElSayed et al. [66] introduced a hybrid intrusion detection model within SDN, combining CNN and random forest. They also introduced a novel regularization technique aimed at enhancing intrusion detection performance. The reported results demonstrated a substantial improvement in detection accuracy, achieving a 97% increase with the hybrid model.

Iqbal et al. [67] introduce an intrusion detection tree (IntruDTree) security model that prioritizes essential security features and constructs a tree-based intrusion detection model using these critical features. This model is effective in predicting unseen test cases
and reduces computational complexity by reducing the number of features. The model’s effectiveness is evaluated through experiments on cybersecurity datasets, assessing metrics like precision, recall, F-score, accuracy, and ROC values. Additionally, the IntruDTTree model is compared with traditional machine learning methods like naive Bayes, logistic regression, support vector machines, and k-nearest neighbor to assess its overall effectiveness in enhancing security. Authors in [68] describe a proposed intrusion detection system (IDS) consisting of two stages. In the first stage, data is collected through dedicated sniffers (DSs) to create a collaborative communication index (CCI), which is periodically sent to a super node (S.N.). In the second stage, the S.N. employs linear regression to analyze the collected CCIs from different DSs to distinguish between benign and malicious network nodes. The paper presents detection characterization for various extreme network scenarios involving power levels and node velocities using two mobility models: random waypoint (RWP) and Gauss Markov (G.M.). The malicious activities studied include blackhole and distributed denial of service (DDoS) attacks. The results indicate high detection rates of over 98% for scenarios with high power levels and node velocities. In comparison, these rates drop to approximately 90% for scenarios with low power levels and node velocities.

Nasir et al. [69] introduce DF-IDS, a model for detecting intrusions in IoT traffic. DF-IDS consists of two main phases: In the first phase, it selects the best features from the feature matrix by comparing various feature selection techniques like Spider Monkey (S.M.), principal component analysis (PCA), information gain (I.G.), and correlation attribute evaluation (CAE). In the second phase, these selected features and assigned labels are used to train a deep neural network for intrusion detection. DF-IDS achieves an impressive accuracy of 99.23% and an F1-score of 99.27%. It outperforms other comparative models and existing studies in terms of accuracy and the F1 score, indicating significant improvements in intrusion detection performance.

Althobaiti et al. [70] suggested an intelligent cognitive computing-driven intrusion detection system designed for industrial cyber–physical systems (CPS). Their approach encompasses a comprehensive data processing pipeline, encompassing stages like data acquisition, preprocessing, feature selection, classification, and optimization, all aimed at identifying anomalies. The literature review is summarized in Table 2.

### Table 2. Summary of Literature Review.

<table>
<thead>
<tr>
<th>Study</th>
<th>Dataset</th>
<th>Techniques Used</th>
<th>Accuracy Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>[44,45]</td>
<td>NSL-KDD</td>
<td>MLP</td>
<td>85.2–87.8%</td>
</tr>
<tr>
<td>[48]</td>
<td>NSL-KDD</td>
<td>Dense random neural network</td>
<td>88.3%</td>
</tr>
<tr>
<td>[49]</td>
<td>NSL-KDD</td>
<td>RNN</td>
<td>89.5%</td>
</tr>
<tr>
<td>[50]</td>
<td>NSL-KDD</td>
<td>APSO-CNN</td>
<td>88.7</td>
</tr>
<tr>
<td>[51]</td>
<td>N-BaloT</td>
<td>CNN</td>
<td>85.7</td>
</tr>
<tr>
<td>[52–54]</td>
<td>NSL-KDD</td>
<td>CNN</td>
<td>90.5</td>
</tr>
<tr>
<td>[55]</td>
<td>NSL-KDD</td>
<td>ANN</td>
<td>92.5%</td>
</tr>
<tr>
<td>[56]</td>
<td>UNSW-15</td>
<td>RNN</td>
<td>92.3%</td>
</tr>
<tr>
<td>[57]</td>
<td>NSL-KDD</td>
<td>MLP</td>
<td>87.6%</td>
</tr>
<tr>
<td>[58]</td>
<td>NSL-KDD</td>
<td>RNN</td>
<td>88.1–90.3%</td>
</tr>
<tr>
<td>[59]</td>
<td>NSL-KDD</td>
<td>LSTM</td>
<td>88.7%</td>
</tr>
<tr>
<td>[60]</td>
<td>NSL-KDD</td>
<td>LSTM</td>
<td>89.4%</td>
</tr>
<tr>
<td>[61]</td>
<td>ISCX2012</td>
<td>logistic regression</td>
<td>88.7%</td>
</tr>
<tr>
<td>[62]</td>
<td>NSL-KDD</td>
<td>LSTM</td>
<td>89–93%</td>
</tr>
<tr>
<td>[63]</td>
<td>NSL-KDD</td>
<td>Random forest</td>
<td>84.5%</td>
</tr>
<tr>
<td>[64]</td>
<td>NSL-KDD</td>
<td>SVM and selective logging</td>
<td>87.7–89.6%</td>
</tr>
<tr>
<td>[65]</td>
<td>Authors generated their dataset using network traffic in a controlled testing environment</td>
<td>SVM</td>
<td>93%</td>
</tr>
<tr>
<td>[66]</td>
<td>NSL-KDD</td>
<td>CNN and random forest</td>
<td>97%</td>
</tr>
<tr>
<td>[70]</td>
<td>NSL-KDD</td>
<td>LSTM</td>
<td>93%</td>
</tr>
</tbody>
</table>

The research project’s implementation will be carried out through a well-structured and systematic methodology to create an advanced IoT security model to enhance the accuracy of security threat detection.

This model, illustrated in Figure 3, embodies a multi-stage approach designed to address IoT security’s intricacies comprehensively.

![Figure 3. Architecture model.](image)

4.1. Step 1: Data Preprocessing

The initial phase of our methodology focuses on preparing the input data for subsequent analysis. This involves a series of crucial sub-steps as shown in Figure 4:

- **Data cleaning**: In this stage, we meticulously examine the datasets, eliminating duplicate entries, addressing missing values, and removing irrelevant or redundant information. This process ensures the dataset’s integrity and quality.
- **Data encoding**: IoT datasets often contain categorical variables that require transformation into a numerical format for machine learning models. We employ appropriate encoding techniques to convert these categorical features into a format suitable for deep learning.
- **Data scaling**: To bring uniformity to the dataset, we apply data scaling techniques, such as normalization or standardization, to ensure that all features have comparable scales. This step facilitates the model’s convergence during training.

![Figure 4. Data cleaning stages.](image)
4.2. Step 2: Data Balancing

Imbalanced datasets are a common challenge in intrusion detection. To mitigate this issue, we employ the synthetic minority over-sampling technique (SMOTE), which generates synthetic samples for the minority class. This balancing method helps prevent the model from being biased towards the majority class, thus improving its ability to detect security threats effectively.

- Compute the imbalance ratio (I.R.):

\[
IR = \frac{\text{Number of Sample of majority class } D_1}{\text{Number of Sample of minority class } D_2}
\]

- Generate synthetic samples for \( D_2 \) Using SMOTE (Synthetic Minority Over-sampling Technique).

\[
x_s = x + \lambda \times (x_{nn} - x)
\]

\( x_s \) is the synthetic sample.
\( x \) is the original sample from \( D_2 \)
\( x_{nn} \) is a randomly selected nearest neighbor of from \( D_2 \)
\( \lambda \) is a random value between 0 and 1 that controls the interpolation.

4.3. Step 3: Feature Selection

Effective feature selection is pivotal for optimizing model performance and reducing computational complexity. We employ a feature selection algorithm (S.A.) tailored to our specific dataset to achieve this. S.A. assists in identifying and retaining the most informative features while discarding irrelevant or redundant ones, resulting in a streamlined and effective feature set.

4.4. Step 4: Classification Using Deep Learning Models

In the subsequent stages, we employ a diverse ensemble of deep learning architectures to perform the classification task, leveraging the unique strengths of each model:

- Convolutional neural networks (CNN): CNNs are adept at capturing spatial and structural patterns within data, making them particularly suitable for image-based intrusion detection or situations where spatial relationships are crucial. CNNs are particularly effective in capturing spatial patterns within data, making them well-suited for image analysis and feature extraction. In intrusion detection, they can be applied to analyze network traffic data and packet payloads. CNNs excel in recognizing spatial relationships and patterns in such data, which can indicate malicious activity. They can detect irregularities or patterns that might be hard to uncover using traditional methods. By including CNNs in the ensemble, the model becomes proficient at recognizing spatial anomalies in network traffic, contributing to the overall robustness of intrusion detection.

- Long short-term memory (LSTM): LSTMs excel in handling sequential data and are well-suited for capturing temporal dependencies in IoT security datasets. They are effective in recognizing patterns that evolve. LSTMs can learn long-term dependencies in data, ensuring that patterns spanning multiple time steps are not overlooked. Their inclusion in the ensemble deep learning framework provides the ability to capture temporal anomalies and identify sophisticated intrusion patterns that might evade detection through more traditional means.

- Gated recurrent unit (GRU): GRUs are a variation of LSTM and balance computational efficiency and performance in sequence modeling tasks, making them a valuable addition to our ensemble. GRUs, like LSTMs, excel in capturing temporal dependencies. By introducing GRUs into the ensemble, the model benefits from a balance of computational efficiency and the capability to detect sequential anomalies, contributing to the model’s ability to detect intrusions resource-efficiently.
The strength of using this ensemble approach lies in the combined power of these models. CNN, LSTM, and GRUs each focus on different aspects of the data: spatial patterns, temporal dependencies, and efficient sequential analysis, respectively. By integrating them and using a voting mechanism, the ensemble can collectively identify anomalies and intrusions more comprehensively. The diverse perspectives provided by these models help reduce false positives and false negatives, enhancing the overall performance of intrusion detection. The voting policy allows the models to reach a consensus, minimizing the likelihood of misclassification and increasing the system’s overall accuracy. This way, the ensemble approach harnesses the strengths of these individual models to create a more robust and effective intrusion detection system.

4.5. Step 5: Training, Testing, and Evaluation

Rigorous testing and evaluation procedures are undertaken following the configuration and training of each deep learning model within the ensemble. This involves partitioning the dataset into training, validation, and test sets, training the models on the training data, and assessing their performance on the test set. Performance metrics such as accuracy, precision, recall, F1-score, and false alarm are employed to quantify the model’s effectiveness in detecting security threats.

This research project adopts a structured approach encompassing data preprocessing, balancing, feature selection, and classification using diverse deep-learning models. The aim is to create an IoT security model that enhances threat detection accuracy and provides a robust and adaptable solution for safeguarding IoT environments.

5. Experimental Evaluation

The proposed scheme has been successfully implemented within the Anaconda environment, a comprehensive open-source platform for deep learning applications [34]. This environment ensures a seamless end-to-end experience for developing and executing our model.

In the upcoming sections, we will delve deeper into our methodology. Section 5.1 will offer an insightful overview of the datasets utilized in our research, setting the foundation for our analysis. In Section 5.2, we will delve into the comprehensive performance measures, including the evaluation metrics and the insightful result analysis, providing a thorough understanding of our model’s effectiveness. Furthermore, the confusion matrix will be presented as an essential tool for visualizing model performance in Section 5.2.

5.1. Dataset

The KDD’99 dataset, curated initially by DARPA in 1999, was assembled using network traffic data recorded in 1998. This dataset has undergone extensive preprocessing, resulting in a representation featuring 41 distinct features per network connection. These features in the KDD’99 dataset are systematically categorized into four groups, each serving a specific purpose: basic features (#1 to #9), content features (#10 to #22), time-based traffic features (#23 to #31), and host-based traffic features (#32 to #41), all thoughtfully outlined in Table 3.

With a voluminous repository of 4,898,430 records, the KDD’99 dataset notably surpasses many other datasets in terms of scale. It is worth noting that within this dataset, there are four primary categories of network attacks, each detailed in Table 2: denial of service (DoS), remote-to-local (R2L, involving unauthorized access from a remote machine), user-to-root (U2R, encompassing unauthorized access to the root), and probe attacks. Using this dataset, researchers and data scientists have extensively leveraged various data mining techniques to detect intrusions in network traffic.

However, two crucial issues were uncovered through statistical analysis, which profoundly impacts the performance of intrusion detection systems applied to the KDD’99 dataset. The most significant among these issues is a substantial number of replicated records. It was observed that approximately 78% and 75% of records in the training and test datasets, respectively, are duplicates. This prevalence of replicated records can inad-
vertently bias learning algorithms, leading them to focus disproportionately on frequent records while neglecting infrequent ones. This oversight can be particularly concerning, as these less frequent records may represent harmful intrusions, such as U2R or R2L attacks.

Despite these challenges, the KDD’99 dataset remains a valuable resource and is still considered an effective benchmark dataset. It plays a pivotal role in facilitating the comparison of various intrusion detection methods by researchers in the field. Furthermore, the dataset’s substantial number of records in both the training and test sets presents a distinct advantage. Researchers can conduct experiments using the complete dataset without random selection, ensuring that evaluation results across different research endeavors remain consistent and comparable. This consistency enhances the reliability and reproducibility of findings, ultimately advancing the state of intrusion detection research.

### Table 3. List of features of the NSL-KDD dataset.

<table>
<thead>
<tr>
<th>F#</th>
<th>Feature Name</th>
<th>F#</th>
<th>Feature Name</th>
<th>F#</th>
<th>Feature Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>Duration</td>
<td>F15</td>
<td>Su attempted</td>
<td>F29</td>
<td>Same srv rate</td>
</tr>
<tr>
<td>F2</td>
<td>Protocol type</td>
<td>F16</td>
<td>Num root</td>
<td>F30</td>
<td>Diff srv rate</td>
</tr>
<tr>
<td>F3</td>
<td>Service</td>
<td>F17</td>
<td>Num file creations</td>
<td>F31</td>
<td>Srv diff host rate</td>
</tr>
<tr>
<td>F4</td>
<td>Flag</td>
<td>F18</td>
<td>Num shells</td>
<td>F32</td>
<td>Dst host count</td>
</tr>
<tr>
<td>F5</td>
<td>Source bytes</td>
<td>F19</td>
<td>Num access files</td>
<td>F33</td>
<td>Dst host srv count</td>
</tr>
<tr>
<td>F6</td>
<td>Destination bytes</td>
<td>F20</td>
<td>Num outbound cmds</td>
<td>F34</td>
<td>Dst host same srv rate</td>
</tr>
<tr>
<td>F7</td>
<td>Land</td>
<td>F21</td>
<td>Is host login</td>
<td>F35</td>
<td>Dst host diff srv rate</td>
</tr>
<tr>
<td>F8</td>
<td>Wrong fragment</td>
<td>F22</td>
<td>Is guest login</td>
<td>F36</td>
<td>Dst host same src port rate</td>
</tr>
<tr>
<td>F9</td>
<td>Urgent</td>
<td>F23</td>
<td>Count</td>
<td>F37</td>
<td>Dst host ssvr diff host rate</td>
</tr>
<tr>
<td>F10</td>
<td>Hot</td>
<td>F24</td>
<td>Srv count</td>
<td>F38</td>
<td>Dst host server rate</td>
</tr>
<tr>
<td>F11</td>
<td>Number failed logins</td>
<td>F25</td>
<td>Srvr error</td>
<td>F39</td>
<td>Dst host svr server rate</td>
</tr>
<tr>
<td>F12</td>
<td>Logged in</td>
<td>F26</td>
<td>Srv servr rate</td>
<td>F40</td>
<td>Dst host rror rate</td>
</tr>
<tr>
<td>F13</td>
<td>Num compromised</td>
<td>F27</td>
<td>Rror rate</td>
<td>F41</td>
<td>Dst host svr rror rate</td>
</tr>
<tr>
<td>F14</td>
<td>Root shell</td>
<td>F28</td>
<td>Srv rror rate</td>
<td>F42</td>
<td>Class label</td>
</tr>
</tbody>
</table>

### 5.2. Performance Measures

Assessing the proposed model’s performance through testing is crucial for gaining insight into its capabilities and grasping its strengths and limitations more comprehensively. In this research, we executed model testing using a five-fold cross-validation method and a testing dataset of around 4000 samples. We then systematically evaluated the model’s performance, utilizing established evaluation metrics throughout the training, validation, and testing phases. Figure 5 summarises the customary performance evaluation criteria employed in this study.

**Figure 5.** Confusion matrix.

The proposed model is assessed using a set of performance metrics derived from the confusion matrix.

\[
\text{Recall} = \frac{TP}{TP + FN}
\]
Precision = \frac{TP}{TP + FP} \\
Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \\
f\text{-measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}

5.3. Result and Discussion

The research introduces a methodology that assesses hard-voting and soft-voting ensemble techniques. These two approaches use three different deep learning algorithms: CNN, GRU, and LSTM, as outlined in Table 4.

<table>
<thead>
<tr>
<th>Methodology</th>
<th>Classifier 1</th>
<th>Classifier 2</th>
<th>Classifier 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hard voting</td>
<td>CNN</td>
<td>GRU</td>
<td>LSTM</td>
</tr>
<tr>
<td>Soft voting</td>
<td>CNN</td>
<td>GRU</td>
<td>LSTM</td>
</tr>
</tbody>
</table>

The study conducted individual evaluations for each of the three classifiers to emphasize the variations in performance among these classifiers and showcase the enhancements achieved through the ensemble model in terms of accuracy. For a more comprehensive understanding of the processes undertaken, Figure 6 describes the steps involved.

Python version 3.8 was employed to create the algorithms, and we utilized the sci-kit to learn the framework in conjunction with the imblearn framework to develop the proposed algorithms. The imblearn framework was particularly useful for handling resampling tasks on the imbalanced dataset.

The study evaluates the accuracy of each algorithm introduced in Table 5, as well as the ensemble methods (hard and soft voting), using the accuracy metric. The accuracy metric is calculated as the number of correctly classified data instances by each algorithm divided by the total samples in the dataset. The study conducted individual evaluations for each of the algorithms to determine their accuracy and showcase the enhancements achieved through the ensemble model in terms of accuracy.

Valuable insights can be gleaned based on the findings depicted in Figure 7 and detailed in Table 5. The ensemble algorithms’ performance metrics show great promise in contrast to the individual classifiers. To be more specific, the hard voting model demonstrated superior accuracy across various dataset variations, including the initial dataset, oversampled dataset, and under-sampled dataset, when compared to the standalone classifiers on the same datasets. Furthermore, the soft voting model outperformed the individual classifiers in both the initial and under-sampled datasets. Notably, when working with the oversampled dataset, the hard voting ensemble model achieved the highest accuracy, whereas the GRU algorithm implementation observed the lowest accuracy.

In Table 6, we compare the top-performing outcomes from our examination of ensemble deep learning classifiers with those of traditional deep learning techniques. The CNN-LSTM and CNN-GRU models emerge as the frontrunners, as indicated in the table. These models achieved remarkable accuracies of 99.7% and 99.6%, respectively, along with corresponding outstanding F1 scores of 0.998 and 0.997. The LSTM-only model follows closely, boasting an overall accuracy of 95.4% and an F1-score of 0.923. The CNN-only model follows with an accuracy of 94.7% and an F1-score of 0.858.
Table 4. Hard voting and soft voting ensemble techniques.

<table>
<thead>
<tr>
<th>Methodology</th>
<th>Classifier 1</th>
<th>Classifier 2</th>
<th>Classifier 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hard Voting</td>
<td>CNN</td>
<td>GRU</td>
<td>LSTM</td>
</tr>
<tr>
<td>Soft Voting</td>
<td>CNN</td>
<td>GRU</td>
<td>LSTM</td>
</tr>
</tbody>
</table>

The study conducted individual evaluations for each of the three classifiers to emphasize the variations in performance among these classifiers and showcase the enhancements achieved through the ensemble model in terms of accuracy. For a more comprehensive understanding of the processes undertaken, Figure 6 describes the steps involved.

Figure 6. Methodology steps for the proposed algorithm.

Table 5. Accuracy of each algorithm with hard and soft voting.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CNN</th>
<th>GRU</th>
<th>LSTM</th>
<th>Hard Voting Ensemble</th>
<th>Soft Voting Ensemble</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>0.947867</td>
<td>0.909091</td>
<td>0.95498</td>
<td>0.956</td>
<td>0.966</td>
</tr>
<tr>
<td>Under Sampling</td>
<td>0.933</td>
<td>0.91</td>
<td>0.943</td>
<td>0.966</td>
<td>0.966</td>
</tr>
<tr>
<td>Oversampling</td>
<td>0.939</td>
<td>0.905</td>
<td>0.945</td>
<td>0.966</td>
<td>0.966</td>
</tr>
</tbody>
</table>

These findings imply that when employing static-based features for IoT intrusion detection, ensemble deep learning models have the potential to surpass the capabilities of conventional deep learning classifiers.

Our research includes a thorough evaluation of two ensemble models, and we have meticulously compared their experimental results with the current state-of-the-art approaches. As depicted in Table 7, our novel methodology demonstrates its effectiveness when contrasted with the existing state-of-the-art methods. Specifically, when applied to the NSL-KDD dataset, our approach not only outperforms its competitors but also attains the highest level of accuracy among all tested methodologies.
The metric is calculated as the number of correctly classified data instances by each algorithm divided by the total samples in the dataset. The study conducted individual evaluations for each of the algorithms to determine their accuracy and showcase the enhancements achieved through the ensemble model in terms of accuracy.

Valuable insights can be gleaned based on the findings depicted in Figure 7 and detailed in Table 5. The ensemble algorithms' performance metrics show great promise in contrast to the individual classifiers. To be more specific, the hard voting model demonstrated superior accuracy across various dataset variations, including the initial dataset, oversampled dataset, and under-sampled dataset, when compared to the standalone classifiers on the same datasets. Furthermore, the soft voting model outperformed the individual classifiers in both the initial and under-sampled datasets. Notably, when working with the oversampled dataset, the hard voting ensemble model achieved the highest accuracy, whereas the GRU algorithm implementation observed the lowest accuracy.

Table 5. Accuracy of each algorithm with hard and soft voting.

<table>
<thead>
<tr>
<th>Deep Learning Algorithms</th>
<th>Dataset</th>
<th>Hard Voting Ensemble</th>
<th>Soft Voting Ensemble</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td>Normal</td>
<td>0.947867</td>
<td>0.956</td>
</tr>
<tr>
<td>CNN</td>
<td>Under Sampling</td>
<td>0.933</td>
<td>0.966</td>
</tr>
<tr>
<td>CNN</td>
<td>Oversampling</td>
<td>0.939</td>
<td>0.966</td>
</tr>
<tr>
<td>GRU</td>
<td>Normal</td>
<td>0.909091</td>
<td>0.966</td>
</tr>
<tr>
<td>GRU</td>
<td>Under Sampling</td>
<td>0.91</td>
<td>0.966</td>
</tr>
<tr>
<td>GRU</td>
<td>Oversampling</td>
<td>0.905</td>
<td>0.966</td>
</tr>
<tr>
<td>LSTM</td>
<td>Normal</td>
<td>0.95498</td>
<td>0.966</td>
</tr>
<tr>
<td>LSTM</td>
<td>Under Sampling</td>
<td>0.943</td>
<td>0.966</td>
</tr>
<tr>
<td>LSTM</td>
<td>Oversampling</td>
<td>0.945</td>
<td>0.966</td>
</tr>
</tbody>
</table>

Figure 7. Experiment results for accuracy.

Table 6. Result experiment for accuracy, precision, recall, and F1 score.

<table>
<thead>
<tr>
<th>Techniques Used</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN → LSTM</td>
<td>0.99734</td>
<td>0.997009</td>
<td>0.999001</td>
<td>0.998004</td>
</tr>
<tr>
<td>CNN → GRU</td>
<td>0.996016</td>
<td>0.995025</td>
<td>0.999001</td>
<td>0.997009</td>
</tr>
<tr>
<td>CNN</td>
<td>0.947867</td>
<td>0.970874</td>
<td>0.769231</td>
<td>0.858369</td>
</tr>
<tr>
<td>GRU</td>
<td>0.909091</td>
<td>0.909091</td>
<td>0.952381</td>
<td>0.930233</td>
</tr>
<tr>
<td>LSTM</td>
<td>0.95498</td>
<td>0.985222</td>
<td>0.869565</td>
<td>0.923788</td>
</tr>
</tbody>
</table>

Table 7. Model comparison with other state-of-the-art methods.

<table>
<thead>
<tr>
<th>Study</th>
<th>Techniques Used</th>
<th>Accuracy Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>[44,45]</td>
<td>MLP</td>
<td>85.2–87.8%</td>
</tr>
<tr>
<td>[48]</td>
<td>Dense random neural network</td>
<td>88.3%</td>
</tr>
<tr>
<td>[49]</td>
<td>RNN</td>
<td>89.5%</td>
</tr>
<tr>
<td>[50]</td>
<td>APSO-CNN</td>
<td>88.7</td>
</tr>
<tr>
<td>[52–54]</td>
<td>CNN</td>
<td>90.5</td>
</tr>
<tr>
<td>[55]</td>
<td>ANN</td>
<td>92.5%</td>
</tr>
<tr>
<td>[57]</td>
<td>MLP</td>
<td>87.6%</td>
</tr>
<tr>
<td>[58]</td>
<td>RNN</td>
<td>88.1–90.3%</td>
</tr>
<tr>
<td>[59]</td>
<td>LSTM</td>
<td>88.7%</td>
</tr>
<tr>
<td>[60]</td>
<td>LSTM</td>
<td>89.4%</td>
</tr>
<tr>
<td>[62]</td>
<td>LSTM</td>
<td>89–93%</td>
</tr>
<tr>
<td>[63]</td>
<td>Random forest</td>
<td>84.5%</td>
</tr>
<tr>
<td>[66]</td>
<td>CNN and random forest</td>
<td>97%</td>
</tr>
<tr>
<td>[70]</td>
<td>LSTM</td>
<td>93%</td>
</tr>
<tr>
<td>Proposed model 1</td>
<td>CNN → LSTM</td>
<td>99.7%</td>
</tr>
<tr>
<td>Proposed model 2</td>
<td>CNN → GRU</td>
<td>99.6%</td>
</tr>
</tbody>
</table>

6. Conclusions

In recent years, the significance of detecting anomalies and malicious attacks in IoT has surged. As the frequency of such attacks continues to rise, the need for robust tools capable of swiftly and accurately identifying intrusions has become paramount. In this research, we introduce an innovative approach leveraging ensemble-based deep learning models for the rapid and precise detection of intrusions in IoT environments while minimizing false positives and negatives. Our proposed models harness the power of three distinct deep learning models: CNN, GRU, and LSTM, each offering unique classification strengths.
These models are thoroughly evaluated using the NSL-KDD open-source dataset, and their performance is benchmarked against standalone models used within the ensemble. Additionally, we compare our results with previous research that employed the NSL-KDD network dataset. Experimental findings unequivocally demonstrate that our proposed model achieves exceptional scores across critical metrics, including accuracy, precision, recall, and F1-Score. In this research, an array of deep learning models encompassing convolutional neural networks (CNNs), long short-term memory (LSTM), and gated recurrent units (GRUs) are harnessed. An innovative ensemble deep learning framework is also introduced, incorporating a voting mechanism within the model’s architecture. This unique approach streamlines the computation and acquisition of hierarchical patterns in the data. Our comprehensive analysis involved comparing the performance of these ensemble deep learning classifiers and traditional deep learning techniques. Notably, the standout models were CNN-LSTM and CNN-GRU, both achieving remarkable accuracy rates of 99.7% and 99.6%, coupled with exceptional F1 scores of 0.998 and 0.997, respectively.

**Author Contributions:** Conceptualization, A.O.; methodology A.O.; software, A.A.T.; validation, A.A.T. and A.O.; formal analysis, A.A.T. and A.O.; investigation, A.A.T. and A.O.; resources, A.O.; data curation, A.A.T.; writing—original draft preparation, A.A.T. and A.O.; writing—review and editing, A.A.T. and A.O.; visualization, A.A.T. and A.O.; supervision, A.A.T.; project administration, All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The authors sincerely acknowledge the Princess Sumaya University for Technology for supporting steps of this work.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**


16. Qin, J.; Wang, J.; Lei, T.; Sun, G.; Yue, J.; Wang, W.; Chen, J.; Qian, G. Deep learning-based software and hardware framework for a noncontact inspection platform for grading. Measurement 2023, 211, 112634. [CrossRef]


20. Attkan, A.; Ranga, V. Cyber-physical security for IoT networks: A comprehensive review on traditional, blockchain and artificial intelligence based key-security. Complex Intell. Syst. 2022, 8, 3559–3591. [CrossRef]


34. Alsulami, A.A.; Abu Al-Haia, Q.; Tayeb, A.; Alqahtani, A. An Intrusion Detection and Classification System for IoT Traffic with Improved Data Engineering. Appl. Sci. 2022, 12, 12336. [CrossRef]


**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.