designs

MDPI

Article

A Deep Learning Approach for Securing IoT Infrastructure with Emphasis on Smart Vertical Networks

Manjur Kolhar * and Sultan Mesfer Aldossary

Department of Computer Science, Prince Sattam Bin Abdulaziz University, Wadi ad-Dawaser 11990, Saudi Arabia; s.aldossary@psau.edu.sa
* Correspondence: m.kolhar@psau.edu.sa

Abstract: As a result of the Internet of Things (IoT), smart city infrastructure has been able to advance, enhancing efficiency and enabling remote management. Despite this, this interconnectivity poses significant security and privacy concerns, as cyberthreats are rapidly adapting to exploit IoT vulnerabilities. In order to safeguard privacy and ensure secure IoT operations, robust security strategies are necessary. To detect anomalies effectively, intrusion detection systems (IDSs) must employ sophisticated algorithms capable of handling complex and voluminous datasets. A novel approach to IoT security is presented in this paper, which focuses on safeguarding smart vertical networks (SVNs) integral to sector-specific IoT implementations. It is proposed that a deep learning-based method employing a stacking deep ensemble model be used, selected for its superior performance in managing large datasets and its ability to learn intricate patterns indicative of cyberattacks. Experimental results indicate that the model is exceptionally accurate in identifying cyberthreats, exceeding other models, with a 99.8% detection rate for the ToN-IoT dataset and 99.6% for the InSDN dataset. The paper aims not only to introduce a robust algorithm for IoT security, but also to demonstrate its efficacy through comprehensive testing. We selected a deep learning ensemble model due to its proven track record in similar applications and its ability to maintain the integrity of IoT systems in smart cities.

Keywords: anomalies; smart vertical networks; smart city; security; IoT infrastructure; mobility

Citation: Kolhar, M.; Aldossary, S.M. A Deep Learning Approach for Securing IoT Infrastructure with Emphasis on Smart Vertical Networks. *Designs* **2023**, *7*, 139. https://doi.org/10.3390/designs7060139

Received: 13 August 2023
Revised: 8 September 2023
Accepted: 7 November 2023
Published: 1 December 2023

1. Introduction

It is possible to find intelligent and sustainable solutions to urbanization's problems in "smart cities", which are built on the foundation of data and communication technology. An intelligent city aims to intelligently convey data by integrating diverse systems. There have been many proposals for "smart city" concepts as a means of better utilizing resources and limiting the expansion of metropolitan areas. The rapid proliferation of IoT devices has ushered in a new era of interconnectedness, enabling a wide range of applications that enhance convenience, efficiency, and automation. A wide range of IoT applications have become an integral part of modern life, from smart homes to industrial processes. The vision of smart cities can be realized through the use of ICTs (information communication technologies), particularly the IoT, which is crucial for ensuring efficient operations [1]. In order for IoT devices to exchange data with nearby items, nodes, and cloud-based apps, they need an internet connection. Using IoT devices in the healthcare setting has enabled the intelligent extension of healthcare services by interconnecting patients with clinicians.

Considering that IoT devices are used in smart cities, cyberattacks may be able to gain unauthorized access to citizens' daily lives without their consent, or they may be able to remotely reconfigure gadgets to an unsafe setting [2–4]. According to Symantec, the number of attacks on the IoT platform this year has increased by 600%, as hackers attempt to exploit the interconnected nature of the targeted devices. A number of security concerns are associated with smart city applications. In the first place, smart city apps are not immune to

zero-day attacks, which take advantage of security flaws in various protocols. Secondly, can network-based cyberattacks be intelligently detected in time to prevent disruption of smart city operations? A third problem is that due to resource limitations (e.g., memory), smart city IoT devices are limited in their capacity to protect themselves and transfer gathered data to remote computers. IoT gadgets are not considered in current intrusion detection system (IDS) solutions.

In the cloud, IoT data are stored on powerful computers with a large amount of storage space. Nevertheless, as the number of IoT devices increases, so does the amount of data transmitted from the IoT terminal layer to the cloud, resulting in latency and congestion issues. In order to address these concerns, the concept of fog computing has been developed [5]. Now, a greater share of the computational work sent to the cloud can be distributed among devices in the fog layer. In this way, data storage and transmission can be addressed while reducing energy consumption, network traffic, and latency. It also aims to speed up the response time of IoT-based smart city applications by bringing the processing closer to the edge device. As a result of the fog layer's ability to detect cyber assaults, there are two main benefits [6]. In the first instance, if assaults are detected in the fog layer, the ISP or network administrator can take precautions in order to limit the extent of the damage.

In spite of these challenges, this paper makes several significant contributions to the field of IoT security in smart cities:

1. A novel security framework: in this paper, we propose a comprehensive security framework that makes use of fog computing to alleviate congestion and reduce latency, thereby enhancing the responsiveness of IoT-based applications in smart cities.
2. In this article, we present a method that uses artificial neural networks (ANNs) based on deep learning to detect vulnerabilities and respond to security threats within smart vertical networks (SVNs).
3. Threat detection and response in real time: our approach utilizes deep learning techniques to identify and mitigate real-time threats.
4. We emphasize the adaptability of our deep learning models, demonstrating their integration into existing IoT infrastructures in order to strengthen the security framework holistically.
5. Detailed validation: through rigorous experimentation and case studies, we demonstrate that our proposed method is effective in strengthening the SVN component of IoT infrastructures, which is crucial for the smooth operation of smart cities.

In focusing on smart vertical networks, this research contributes to strengthening the IoT landscape and ensuring its security and reliability. As a result of our research, we advocate for a nuanced approach to IoT security, combining deep learning's predictive power with strategic security measures, as a means of advancing towards a resilient smart city paradigm.

2. Related Works

Researchers have developed different optimization methods to address these threats. A novel approach utilizing machine learning was proposed [7] for forecasting IoT connection activity using machine learning. This technique focuses on monitoring the interaction between services within a distributed multidimensional IoT microservices framework deployed within an IoT environment. The authors intend to correct IoT service forecasting of behaviors solely based on this observation. Within an IoT environment, the technique described in this study focuses on the continuous learning of microservice models. In order to cluster the data, K-means- and BIRCH-based clustering techniques are used [8]. As a result of this particular scenario, the clustering algorithm considers cluster centers that fall within a range of three standard deviations to be part of the same cluster.

Through the incorporation of an online learning communication model, the proposed model aims to enhance the process of cluster formation. According to previous research [9], researchers have proposed a novel defense mechanism called joint trust light probe-based

defense (TLPD). In an industrial IoT setting, the goal of this approach was to identify hostile network nodes responsible for On and Off attacks. An On and Off attack occurs when a node within an IoT network is deliberately targeted when it is either active (On) or inactive (Off). To assist in anomaly detection, a light sample routing method was incorporated into the framework. This mechanism also involves assessing each node's confidence level.

In their study, Ref. [10] proposed a novel approach for detecting anomalies in IoT backbone networks. In order to implement the proposed technique, two stages were involved, namely, dimension reduction and classification. These attacks were of particular interest due to their potential negative consequences. PCA and linear discriminant analysis were used to reduce the dimensionality of the dataset. In order to detect anomalies, they applied naive Bayes and K-nearest neighbor (KNN) algorithms. According to their evaluation of their approach, they were able to identify 84.82% of the candidates. In previous research, Kozik et al. [11] described a new method for detecting assaults. The technique utilized the Apache Spark cloud architecture and the extreme learning machine (ELM) method. Netflow packets can be efficiently and effectively analyzed and processed using the ELM architecture and its associated properties.

In the field of artificial intelligence, machine learning has emerged as a prominent area. This has led to significant success in solving computational challenges in a range of domains. Recent years have seen a dramatic increase in the use of deep networks, including deep learning, extreme learning, and deep extreme neural networks [12–14]. Most novel attacks share certain characteristics with known attacks. Due to their complexity and intricacy, it can be challenging to describe and represent relationships.

Using deep learning techniques, it is possible to model intricate non-linear relationships. The system accomplishes this by learning multiple levels of data representations in order to gain a comprehensive understanding of the data. As a result of these representations, the system is able to capture and comprehend complex patterns and structures within the data at varying levels of abstraction. An increasing amount of attention has been paid to the use of deep neural networks in the detection of threats in social networks. In these networks, multiple layers of non-linear processing units are arranged in a hierarchical structure. As a result of this architecture, deep neural networks are capable of extracting and transforming features, making them a promising approach for detecting attacks on social networks. The purpose of this study is to determine whether or not a specific intervention has an effect on a specific outcome.

To determine whether IoT contexts meet the secrecy and safety requirements, Ref. [15] examines legal concerns and regulatory methods. A number of studies have been performed regarding the security and confidentiality of the widely distributed IoT, including those by [16,17]. They also discussed the benefits of dispersed IoT in terms of security and privacy, along with some challenges. According to a recent survey, ransomware attacks and other security issues are only two examples of the rising risks and vulnerabilities associated with IoT schemes. According to [18], data privacy and security are important considerations in the IoT environment. Additionally, the researchers identified three obstacles to the development and deployment of machine learning in the IoT. There has been considerable research conducted on how to classify security threats associated with the IoT [19,20].

In [21], a cyberattack alert system based on fog–cloud architecture and ensemble learning is proposed. As a result of the ensemble method, a number of models are integrated, such as Decision Trees, Naive Bayes, and Random Forests, developed by the learners. In [22], we see an innovative method for detecting anomaly-based assaults in an unbalanced dataset using an LSTM autoencoder. During training, only samples from typical classes are used. In [23], the focus is on modeling the available characteristics and extracting the most relevant ones. Using deep learning, the researchers sort intrusions into categories. L. Rondon et al. [24] contend that cybersecurity is overlooked or not considered in the design of IoT devices. M. Alajanbi et al. have written a review study highlighting the efficacy of several intrusion detection methods [25]. We summarize in Table 1 the key contributions of the existing literature on IoT security and their implications for smart

cities. As a result of association rule mining and artificial neural networks (ANNs), F. Safara et al. [26] developed a system for identifying intrusions in communication networks. Additionally, the proposed method is based on ANNs. However, it has been developed with the intention of detecting intrusions in Internet of Things networks. It has been demonstrated that deep learning algorithms can be utilized in IoT security by M. Abdel-Basset et al. In addition, they discussed the use of deep learning in the context of IoT privacy. The creation of rule-based intrusion detection systems tailored to a particular network, however, is more straightforward. As long as they are used in the proper system, they perform exceptionally well. The knowledge-driven, rule-dependent, system collapse attack intrusion prevention system was developed by [27,28].

Table 1. Comparative analysis of literature on IoT security: a synopsis of advantages, disadvantages, and contributions to smart city applications.

Ref.	Authors and Year	Main Contribution	Advantages	Disadvantages	Applicability to IoT Security
[7]	Saba et al., 2022	Anomaly-based IDS for IoT networks via DL	High detection accuracy, adaptability to new threats	Requires substantial training data, may have longer training times	Highly applicable for network security
[10]	Guo et al., 2023	Traffic management in IoT networks with SDN orchestration	Efficient traffic handling, enhanced through GNN and MAB	Complexity in deployment and maintenance, possible overhead	Applicable for managing network traffic
[16]	Patel and Patel, 2022	Blockchain for IoT healthcare data confidentiality	Enhanced data integrity and access control	Blockchain complexity, scalability issues	Applicable for secure healthcare data
[22]	Elsayed et al., 2020	Network anomaly detection using LSTM-based autoencoder	Effective for sequential data, captures temporal dependencies	Can be computationally intensive, challenging to tune	Applicable for continuous data streams
[23]	Elsayed M et al., 2020	Improved CNN for IDS in SDNs	Higher accuracy in detection with an improved learning approach	Specific to SDNs, may not generalize to other network types	Applicable for intrusion detection in SDNs
[14]	Sriranjani et al., 2023	Machine learning for detecting replay attacks in smart grid	Effective in identifying specific replay attack patterns	Limited to smart grid environments	Applicable for smart grid security
[18]	Rao and Deebak, 2022	Security and privacy in smart cities/industries	Comprehensive review of technologies and applications	No new model or algorithm proposed	Provides a framework for addressing smart city security challenges
[21]	Kumar et al., 2021	Cyberattack detection for IoMT networks	Uses ensemble learning and fog–cloud architecture for efficiency	Requires specific infrastructure setup	Highly applicable for IoMT security

3. Database Description

The effectiveness of an IDS is dependent on the data used for training. The inability to test detection algorithms on a current real-world dataset is a major roadblock. Privacy and legal concerns are the primary reasons why public datasets are unavailable in the intrusion

detection field. Two of the most current practical datasets, ToN-IoT and InSDN, are used to train and test the proposed method.

3.1. ToN-IoT Dataset

For the ToN-IoT dataset, researchers from the Australian Defense Force Academy (ADFA) turned to the Cyber IoT Lab at the School of Engineering and Information Technology (SEIT) at UNSW Canberra [29]. The dataset was assembled via machine learning, and it contains many different types of cyberattack and non-attack events from the Internet of Things systems. To replicate the performance and scalability of the automotive IoT and Enterprise 4.0 systems through the integration of several digital and physical resources, as well as hacking infrastructure, cloud and fog environments, and IoT sensors, a new testbed was built at the IoT laboratory. The latest distributed denial of service (DDoS) and ransomware attacks against smart cities are included in this collection. Attacks on the IoT network target web apps, IoT gateways, and other computer systems. There are 43 features in all, and the 461,043 remarks are split between 300,000 "normal" remarks and 161,043 "attack" observations, as shown in Figure 1. In total, 70% of the information was put to good use in the form of instruction, while the remaining 30% was put to the test.

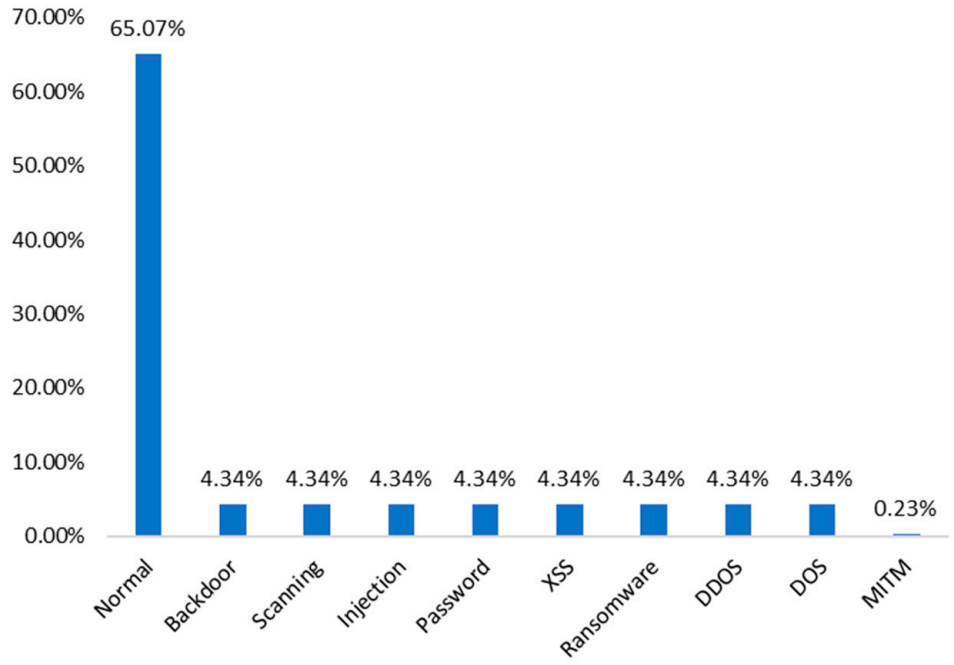


Figure 1. Data distribution of the ToN-IoT dataset.

3.2. InSDN Dataset

Probe, DoS application, online assaults, brute force attack, password speculation, U2R, and DDoS attacks are only a few of the numerous attack types included in the InSDN dataset. Furthermore, there are a number of shared characteristics of InSDN normal traffic. To more accurately reflect the nature of real-world assaults, the dataset includes intrusions from both internal and external networks. CSV formatted data on 80+ metrics, including protocol, duration, bytes, packets, etc. There are a total of 343,939 occurrences in the dataset, consisting of 68,424 occurrences of "normal" traffic and 275,515 occurrences of "attack" traffic, as shown in Figure 2.

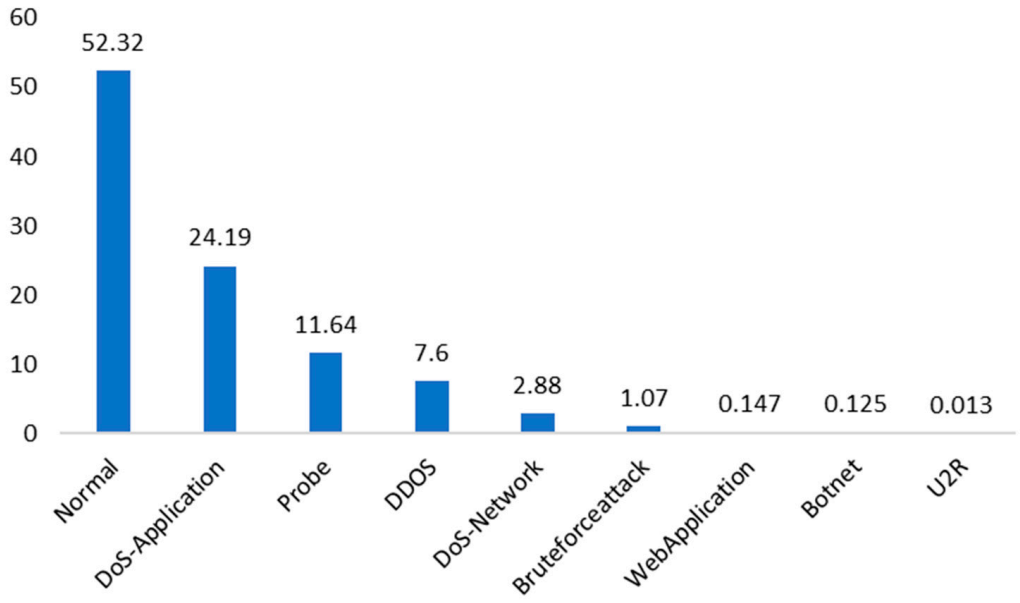


Figure 2. Data distribution of the InSDN dataset.

4. Proposed Methodology

This study presents a comprehensive approach to safeguard the IoT infrastructure within smart cities from cyberthreats, with a special focus on SVN. The overarching processing flow, as shown in Figure 3, encompasses multiple key stages, starting with the acquisition and preparation of a smart city dataset, followed by rigorous pre-processing to ensure data quality and compatibility. A pivotal component of the methodology involves the utilization of stacked ensemble deep learning techniques, a unique and robust approach.

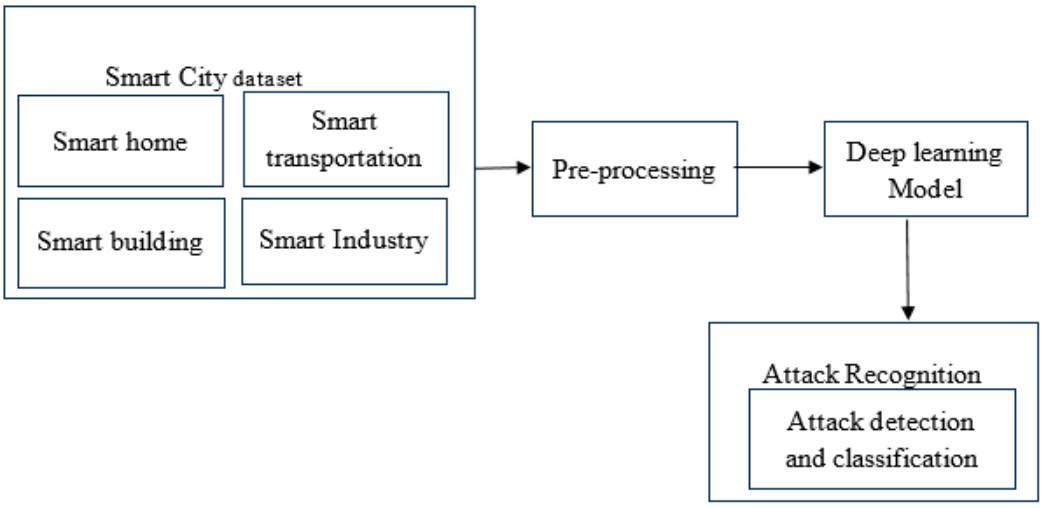


Figure 3. The overall structure of proposed methodology.

The system's core is the deep learning classifier, finely tuned to discern normal network behavior from anomalous activities. Subsequently, the refined system proceeds to attack recognition, adeptly identifying and differentiating various cyberattacks within the IoT ecosystem. The final stage encompasses attack detection and classification, where the system's learned insights are leveraged to swiftly and accurately categorize detected threats. This holistic approach promises to bolster the resilience of smart cities' IoT infrastructure against evolving cyberthreats, explicitly addressing the intricacies of smart vertical networks for a comprehensive and effective defense mechanism.

Stacked Ensemble Model

The proposed deep ensemble model combines the predictive power of the Xception convolutional neural network (CNN) and the bidirectional long short-term memory (Bi-LSTM) model to effectively detect and classify attacks within the IoT infrastructure of smart cities, with a focus on SVN, as shown in Figure 4. This ensemble model harnesses the strengths of both models to enhance accuracy and robustness. The input features for the ensemble model consist of the raw data representing the network behavior, which are carefully pre-processed to extract relevant patterns and features. These features are then fed into the Xception CNN and the Bi-LSTM model for individual attack predictions. The Xception CNN excels in capturing spatial features from the data, leveraging deep convolutions, while the Bi-LSTM focuses on sequential and temporal patterns, adept at modeling the dynamic nature of network activities.

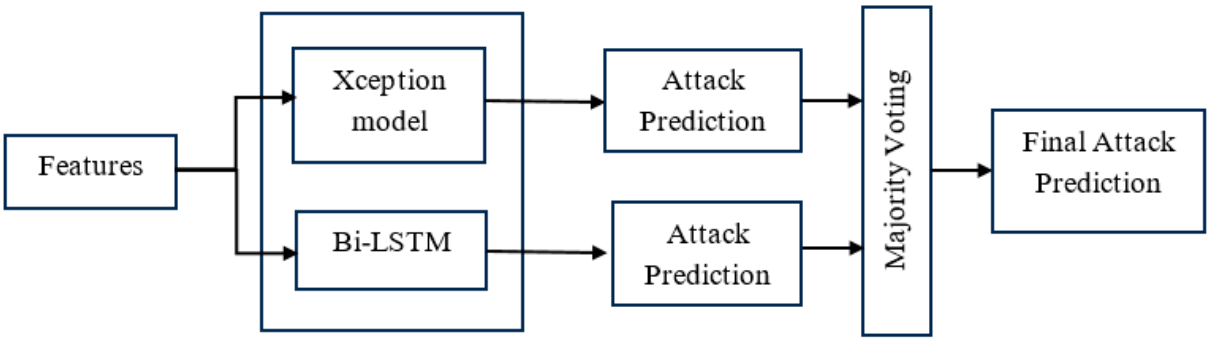


Figure 4. Stacked deep ensemble model (SDEM) for intrusion detection in IoT.

Once the individual attack predictions are obtained from both models, a majority voting mechanism is applied to determine the final attack classification. This approach capitalizes on the collective decision-making process, ensuring that the last forecast is based on the consensus of the Xception and Bi-LSTM models. Through the use of this ensemble strategy, the impact of potential false positives or false negatives from either model may be mitigated, and the accuracy and consistency of attack detection will be improved. By combining the unique capabilities of the Xception CNN and the Bi-LSTM model through a majority voting mechanism, the proposed ensemble approach delivers a robust and comprehensive solution to safeguard the IoT infrastructure of smart cities against cyberattacks, specifically tailored to address the intricacies of smart vertical networks.

Xception CNN Model

The Xception (Extreme Inception) model is a deep CNN architecture that is based on the idea of separating feature learning into two stages: one for spatial features (using depthwise separable convolutions) and another for channel-wise features (using 1×1 convolutions). This separation of feature learning helps the network learn more efficient and discriminative representations. Figure 5 is the basic structure of the Xception model, which consists of repeated blocks of depthwise separable convolutions and 1×1 convolutions. The input of the Xception model is an image or a feature map from a previous layer. The entry flow part of the network performs initial feature extraction and reduction. It usually consists of a series of convolutional, pooling, and activation layers to reduce the spatial dimensions while boosting the channel count. The middle part of the network is where the depthwise separable convolutions are applied, enabling the network to capture spatial features efficiently. It is not unusual for this part of the graph to be composed of repeated blocks of separable convolutions and residual connections. A channel-wise feature is combined using convolutions, and then a global average pooling algorithm and softmax are applied to the network exit flow prior to classification.

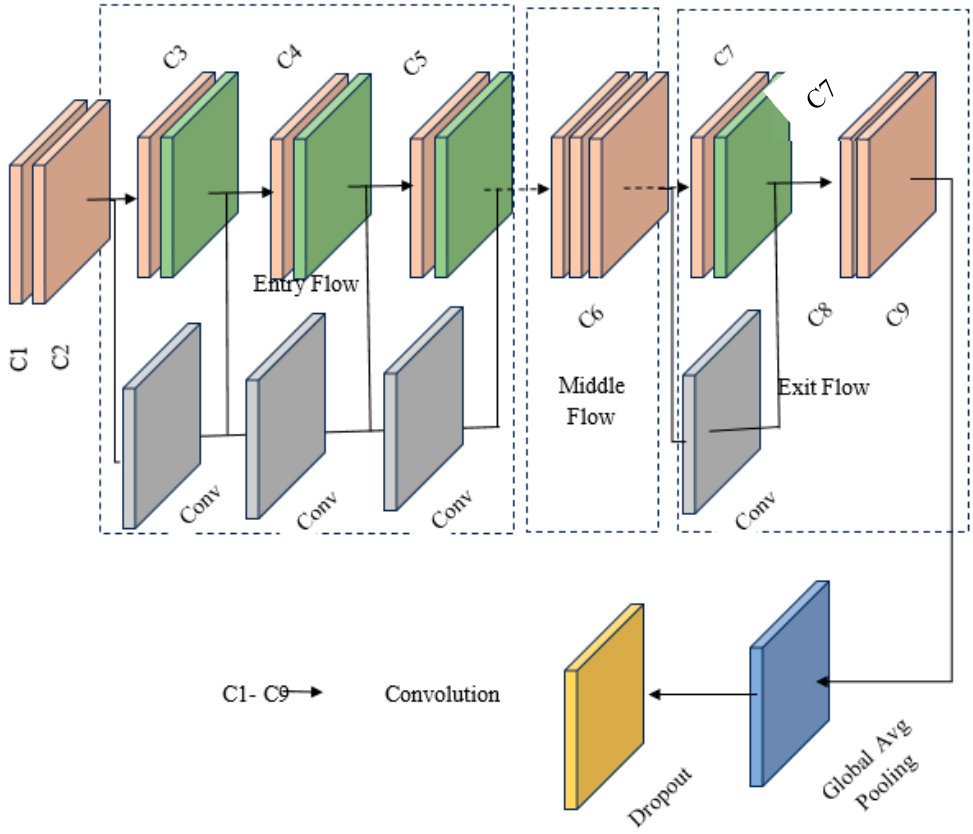


Figure 5. Basic structure of the Xception model.

Entry Flow (C1–C5)

In the entry flow, the input image is first processed. Typically, it consists of the following components:

Convolutional Layers: These are initial standard convolutions (not depth-separable) with increasing numbers of filters, typically beginning with 32 or 64. The purpose of these algorithms is to quickly reduce the spatial dimension of the input. Convolutional Blocks with Depth Separation: After the initial standard convolutions, depthwise separable convolutions are introduced. Each of these layers separates the convolution operation into depthwise spatial convolutions (which act on each input channel separately) and pointwise spatial convolutions (1×1 convolutions that combine the outputs of the depthwise spatial convolutions). Maximum Pooling: Maximum pooling is often used to reduce the spatial dimensions of feature maps between blocks.

Middle Flow (C6)

This flow is the core of the network and consists of a number of identical blocks (e.g., eight). Typically, each block contains the following information: Convolutional Layers with Depth Separation: These layers are the basic building blocks of the middle flow, which can be used to extract features from the input data. Bottlenecks: This involves the use of 1×1 convolutions with a reduced number of filters to compress the feature representation. The middle flow does not alter the dimensions of the feature maps, but instead concentrates on extracting more complex features.

Exit Flow (C7–C9)

Exit flows complete the process and prepare feature maps for final classification. In general, it consists of:

Flows with depth-separable convolutional layers: Similar to the previous flow, but with a greater number of filters. Pooling Layers: It is common to use global average pooling to reduce each feature map to a single value. Fully connected layers: These are used at

the end of the network to facilitate classification. There are typically as many neurons in the last layer as there are classes in the classification task. This process helps in enhancing the representation of the input data by incorporating information from different channels. Mathematically, the depthwise separable convolution operation can be represented as

$$D(x) = \sum W_d \cdot x \quad [for\ each\ channel], \tag{1}$$

And pointwise convolution as

$$P(D(x)) = \sum W_p \cdot D(x), \tag{2}$$

where x is the input feature map (for a single channel), W_d is the depthwise convolution filter, W_p is the pointwise convolution filter, $D(x)$ is the intermediate feature map after depthwise convolution, $P(D(x))$ is the final output after pointwise convolution.

The output of the depthwise convolution has C_{in} feature maps. Mathematically, for a depthwise convolution operation, the output at position $(i,\ j)$ in feature map c is given by the depthwise convolution equation:

$$D(x, y, z) = \sum_{m,n} \left(input(x \cdot S + m, y.S + n, z) \cdot DepthwiseFilter(m, n, z) \right), \tag{3}$$

where (x, y, z) represents the position (x, y) in the c-th channel of the output feature map, S is the stride of the convolution, $DepthwiseFilter(m, n, z)$ is the depthwise filter at position (m, n) in channel c, $input(x \cdot S + m, y.S + n, z)$ is the input value at position $(x \cdot S + m, y.S + n, z)$.

The output of the pointwise convolution has a reduced number of channels (controlled by the number of 1×1 filters). Mathematically, for a pointwise convolution operation, the output at position $(i,\ j, C_{out}\ )$ is given by

$$P(x, y, C_{out}\ ) = \sum_c \left(DW(x, y, c) \cdot PointwiseFilter(c, C_{out}\ ) \right), \tag{4}$$

where (x, y, C_{out}) represents the position (x, y) in the C_{out}-th channel of the output feature map, $PointwiseFilter(c, C_{out})$ is the pointwise filter that combines input channel c to produce output channel C_{out}, $DW(x, y, c)$ is the result of the depthwise convolution at position (x, y) in channel c.

An Xception block typically consists of depthwise separable convolutions, batch normalization, activation functions (e.g., ReLU), and residual connections. The residual connections help with gradient flow and make training deeper networks more stable. Mathematically, an Xception block can be represented as follows:

$$Y = Activation(BN(Depthwise\_Conv(X) + X)), \tag{5}$$

where X is the input feature map, $Depthwise\_Conv$ is the depthwise separable convolution operation, BN is batch normalization, Activation is the activation function (ReLU), Y is the output of the Xception block. This basic structure of the Xception model captures the essence of its design, focusing on efficient feature learning using depthwise separable convolutions and 1×1 convolutions. In practice, the model architecture may have additional complexities, such as multiple Xception blocks, pooling layers, and fully connected layers for classification. The equations for the forward and backward LSTM cells are as follows:

Input Gate:

$$i_t = \sigma(W_{xi} x_t + W_{hi} \mathrm{h}_{t-1} + b_i) \tag{6}$$

Forget Gate:

$$f_t = \sigma\Big(W_{xf} x_t + W_{hf} \mathrm{h}_{t-1} + b_f\Big) \tag{7}$$

Output Gate:

$$o_t = \sigma(W_{xo} x_t + W_{ho} \mathrm{h}_{t-1} + b_o) \tag{8}$$

Candidate Cell:

$$c'_t = tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (9)$$

Cell State:

$$c_t = f_t \odot c_{t-1} + i_t \odot c'_t \quad (10)$$

Hidden State:

$$h_t = o_t \odot \tanh(c_t) \quad (11)$$

In the development of an IoT intrusion detection system, the initial step involves partitioning the dataset into distinct subsets for training, validation, and testing. Through this data division, the Bi-LSTM model is trained using the training subset, fine-tuning its internal parameters such as weights and biases. Validation occurs concurrently, with the model's performance assessed on the validation set to prevent overfitting and optimize its generalization abilities via backpropagation and gradient descent. Subsequent to training, the trained Bi-LSTM is applied to make predictions regarding the nature of network traffic: distinguishing between normal traffic and signs of a potential intrusion. This task primarily involves binary classification, where the model aims to accurately classify instances as either benign or indicative of an intrusion, based on the learned patterns from the training data.

Majority voting in the context of an ensemble model combining an Xception CNN and a Bi-LSTM is a technique used to make predictions based on the consensus of individual models. The goal is to advance the overall accuracy and robustness of intrusion detection in IoT. In the majority voting scheme, both the Xception CNN and the Bi-LSTM models make individual predictions on the same input data. These predictions are then combined through a voting mechanism to determine the final prediction. The group that receives the most votes wins from the two models is chosen as the ensemble's final prediction.

Let us denote the output probabilities from the Xception CNN as $P_{Xception}(c_i)$, where c_i represents the class label for class i. Similarly, let the output probabilities from the Bi-LSTM be denoted as $P_{\text{BiLSTM}}(c_i)$. The majority voting ensemble decision can be represented as follows:

$$P_{Ensemble}(c_i) = \frac{1}{2}\left(P_{Xception}(c_i) + P_{\text{BiLSTM}}(c_i)\right) \quad (12)$$

In this equation, we take the average of the predicted probabilities from both models, effectively giving equal weight to each model's prediction. The class label ci with the maximum $P_{Ensemble}(c_i)$ is selected as the final forecast. This majority voting approach helps to mitigate the weaknesses of each individual model by leveraging their diverse capabilities. As a result, the IoT intrusion detection system functions better as a whole, leading to more reliable and accurate predictions. Xception CNN and Bi-LSTM are combined to take use of their synergies, the ensemble effectively handles both feature extraction from raw data and sequential pattern recognition, improving the overall effectiveness of intrusion detection in IoT environments

Upon completing the intrusion detection phase, thorough evaluation of the system's effectiveness is essential. This evaluation hinges on the use of a dedicated testing dataset, enabling the assessment of the model's performance in real-world scenarios. Commonly employed evaluation metrics, including accuracy, precision, recall, F1-score, and the area under the receiver operating characteristic (ROC) curve (AUC-ROC), provide valuable insight into the model's ability to effectively recognize and discriminate between normal and malicious network behavior, forming a critical aspect of the system's overall robustness assessment.

5. Result and Discussion

The experimental setup for evaluating the proposed in the context of intrusion detection was conducted using MATLAB on a computer system equipped with an Intel Core-i7 processor, 8 GB of RAM, and running the Windows 10 operating platform. The proposed solution implemented on this hardware configuration, and its effectiveness in detecting intrusions, was rigorously tested. Two key evaluation metrics were utilized: the confusion

matrix and the ROC curve. The confusion matrix serves as a structured table commonly used in multiclass assessment scenarios. It provides vital data on the operation of the system, detailing the number of instances correctly and incorrectly classified for each class. This matrix aids in understanding the distribution of predicted classifications compared to the actual ground truth labels. Additionally, the ROC curve offers insight into the system's capacity to tell the difference among favorable and unfavorable situations across various threshold settings. The accuracy is the fraction of dataset samples where the network successfully predicted all of the data points.

$$Accuracy(Ac) = \frac{Tr_{ps} + Tr_{ng}}{Tr_{ps} + Tr_{ng} + Fa_{ps} + Fa_{ng}} \tag{13}$$

The precision of a model to identify attacks is defined as the fraction of false positives that are corrected out of all false positives that the model detects.

$$Precision(Pr) = \frac{Tr_{ps}}{Tr_{ps} + Fa_{ps}} \tag{14}$$

The recall (Rr) is the proportion of successful tests relative to the number of rectified samples detected by the model.

$$\text{Recall(Rr)} = \frac{Tr_{ps}}{Tr_{ps} + Fa_{ng}} \tag{15}$$

The percentage of true negatives (TNRs) identified accurately is the measure of specificity.

$$\text{specificity(Sp)} = \frac{Tr_{ng}}{Fp_{ps} + Tr_{ps}} \tag{16}$$

The F1-score measures the reliability of a test by averaging the percentage of true positives.

$$\text{F1} - score(F1) = 2 \times \frac{Pr \times Rr}{Pr + Rr} \tag{17}$$

According to Table 2, configurations with three LSTM layers and 300 hidden layers showed the highest accuracy of 99.8%, making them promising choices for IoT intrusion detection. The model's performance is significantly affected by the number of LSTM layers and hidden layers. The InSDN dataset is analyzed in Table 3 by a specific model known as the stacked deep ensemble model (SDEM). Based on factors such as the number of LSTM layers and size of hidden layers, the SDEM is evaluated and compared across various configurations. As indicated in subfigures (a) and (b), Figure 6 illustrates the accuracy performance of the SDEM in two distinct scenarios. As a result of the SDEM's accuracy percentages, instances within each dataset are correctly classified. According to subfigure (a), SDEM made 99.8% accurate predictions on the ToN-IoT dataset. It indicates that the model successfully and consistently discriminated between different categories or groups of data instances in the ToN-IoT dataset.

The InSDN dataset, accuracy is 99.6% according to subfigure (b). We are therefore able to classify instances very accurately within the InSDN dataset thanks to our model. The model's predictions had a high level of reliability, indicating that it was capable of identifying a variety of threats in the InSDN dataset. The accuracy percentages provide valuable insight into the robustness and efficiency of SDEMs applied to these specific IoT datasets. This figure illustrates the model's remarkable performance given the demanding task of multiclass classification in cybersecurity and the Internet of Things.

Table 2. Performance analysis of the model comparing the specified configurations of the ToN-IoT dataset.

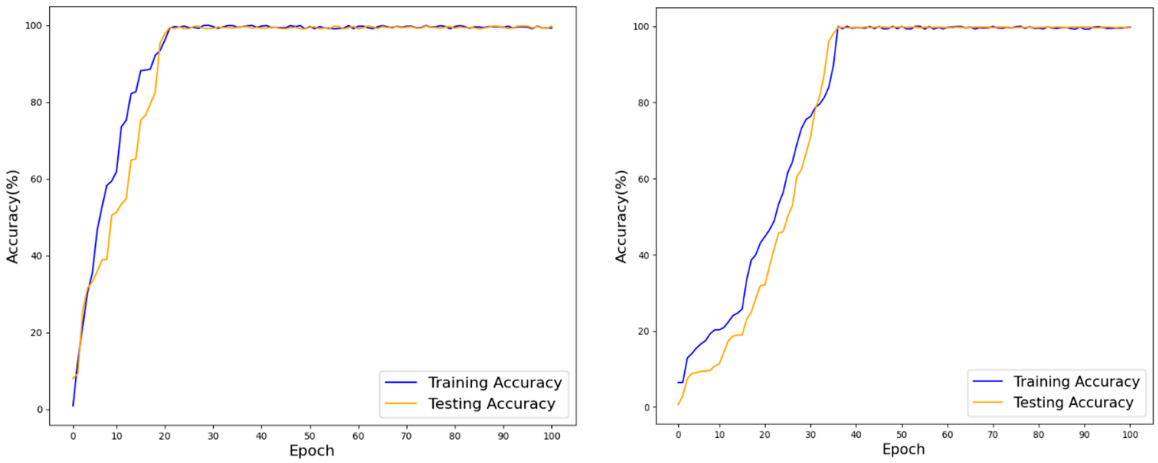
LSTM Layers	Hidden Layers	Accuracy	ROC	Precision	Recall	F1-Score
1	100	98.2%	0.95	0.92	0.89	0.905
1	200	97.6%	0.94	0.91	0.87	0.89
1	300	98.0%	0.95	0.92	0.88	0.9
1	400	97.4%	0.94	0.90	0.86	0.88
2	100	98.4%	0.96	0.93	0.90	0.915
2	200	98.6%	0.97	0.94	0.91	0.925
2	300	98.8%	0.97	0.95	0.92	0.935
2	400	98.7%	0.97	0.94	0.91	0.925
3	100	99.0%	0.98	0.95	0.93	0.94
3	200	99.1%	0.98	0.96	0.93	0.945
3	300	99.8%	0.99	0.98	0.97	0.975
3	400	99.5%	0.98	0.97	0.95	0.96
4	100	98.8%	0.97	0.94	0.92	0.93
4	200	98.9%	0.97	0.95	0.92	0.935
4	300	99.6%	0.98	0.97	0.96	0.965
4	400	99.2%	0.98	0.96	0.94	0.95

Table 3. Performance analysis of the model comparing the specified configurations of the InSDN dataset.

LSTM Layers	Hidden Layers	Accuracy	ROC AUC	Precision	Recall	F1-Score
1	100	95.2%	0.951	0.942	0.962	0.952
1	200	96.5%	0.965	0.954	0.974	0.964
1	300	97.1%	0.971	0.962	0.978	0.970
1	400	96.8%	0.968	0.957	0.975	0.966
2	100	97.5%	0.975	0.968	0.978	0.973
2	200	98.3%	0.983	0.978	0.983	0.980
2	300	98.7%	0.987	0.982	0.987	0.985
2	400	98.5%	0.985	0.980	0.988	0.984
3	100	98.8%	0.988	0.983	0.989	0.986
3	200	99.1%	0.991	0.986	0.992	0.989
3	300	99.8%	0.998	0.996	0.998	0.997
3	400	99.5%	0.995	0.994	0.995	0.995
4	100	99.0%	0.990	0.988	0.991	0.989
4	200	99.2%	0.992	0.990	0.993	0.992
4	300	99.6%	0.996	0.995	0.997	0.996
4	400	99.3%	0.993	0.992	0.994	0.993

This figure illustrates the proposed model's ROC curve analysis for the ToN-IoT dataset. A classification model's effectiveness is assessed by ROC curves. In multiclass situations, they can also be used for binary categorization. In this context, the ROC curve demonstrates how well the SDEM distinguishes between different classes or categories within the ToN-IoT dataset, typically representing a variety of IoT device behaviors.

Figure 7 displays the ROC curve analysis results for the stacked deep ensemble model (SDEM) when applied to the InSDN dataset. The ROC curve in this figure demonstrates how effectively the SDEM differentiates between different classes or types of threats within the InSDN dataset. The ROC curve shows the tradeoff among sensitivity and specificity (the number of false positives) at various cutoffs for making a diagnosis.



(**a**) ToN-IoT dataset (**b**) InSDN dataset

Figure 6. Accuracy of the SDEM applied to the (**a**) ToN-IoT dataset and (**b**) InSDN dataset.

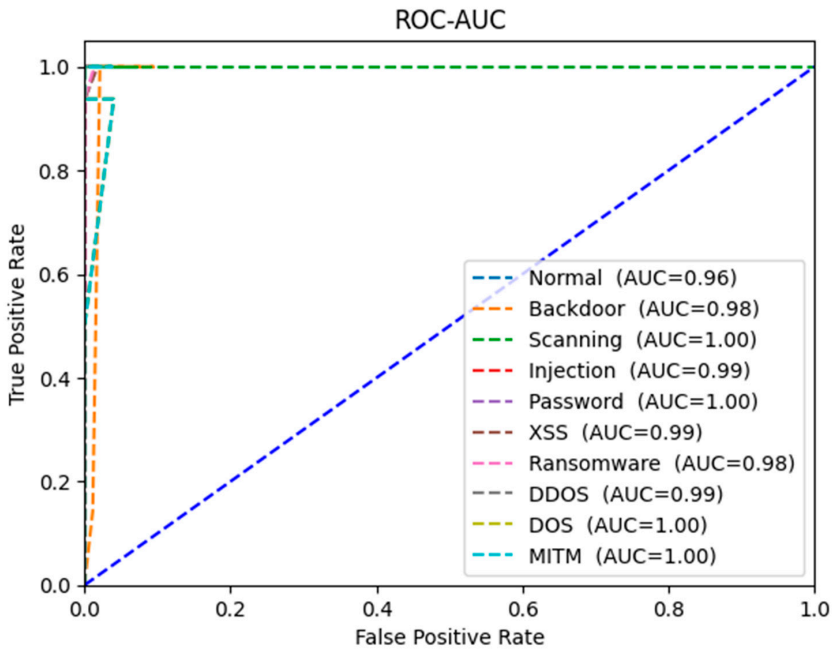


Figure 7. ROC of the proposed model applied to the ToN-IoT dataset.

Figure 8 displays the confusion matrix analysis results specifically applied to the ToN-IoT dataset; a collection of data instances related to the IoT. The confusion matrix is a structured table that provides comprehensive insight into the model's performance in classifying different categories; in this case, various IoT device behaviors categorized as threat types. The provided values within the confusion matrix entries denote the accuracy of classification for each threat type within the ToN-IoT dataset. These values range from 0.98 to 1.00, and are associated with specific threat categories. For instance, a value of 0.99 for the "Normal" class indicates that the model correctly classified normal behavior with a high accuracy of 99%. Similarly, values of 1.00 for categories like "Injection", "XSS", "Ransomware", and "DOS" suggest that the model achieved perfect accuracy in classifying instances from these classes. Additionally, values such as 0.98 for the "DDOS" category and 0.99 for "MITM" reflect slightly lower but still impressive accuracy levels in identifying these specific threat types.

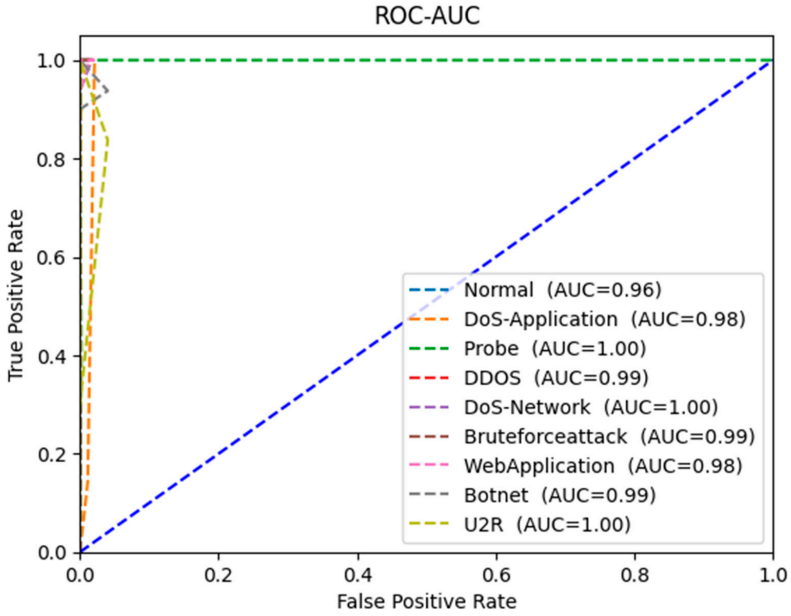


Figure 8. ROC of the SDEM applied to the InSDN dataset.

Figures 9 and 10 presents a confusion matrix for the InSDN dataset that offers a comprehensive view of the performance of a multiclass classification model applied to a diverse set of classes representing different types of network activities or behaviors. Understanding the model's ability to predict may be gleaned from the confusion matrix, specifically in terms of how well it correctly assigns instances to their respective classes, as well as where it might have encountered difficulties. The provided high accuracy rates for some classes (e.g., 'DoS-Application = 1.0', 'DDOS = 1.0', 'Bruteforceattack = 1.0', 'WebApplication = 1.0', 'U2R = 1.0') suggest that the model is remarkably adept at identifying instances from these categories. On the other hand, slightly lower accuracy rates for other classes (e.g., 'Normal = 0.99', 'Probe = 0.99', 'DoS-Network = 0.99', 'Botnet = 0.99') still indicate strong performance, with very few instances being misclassified.

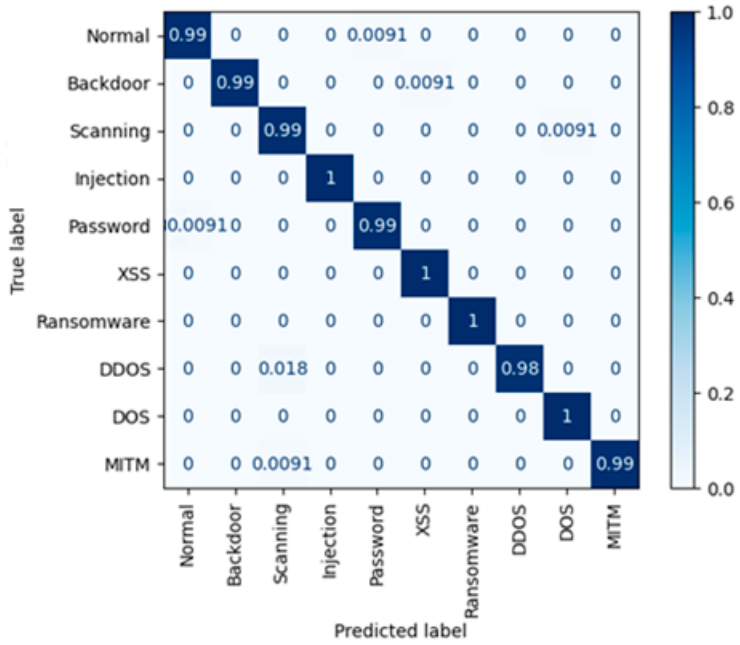


Figure 9. Confusion matrix for the ToN-IoT dataset.

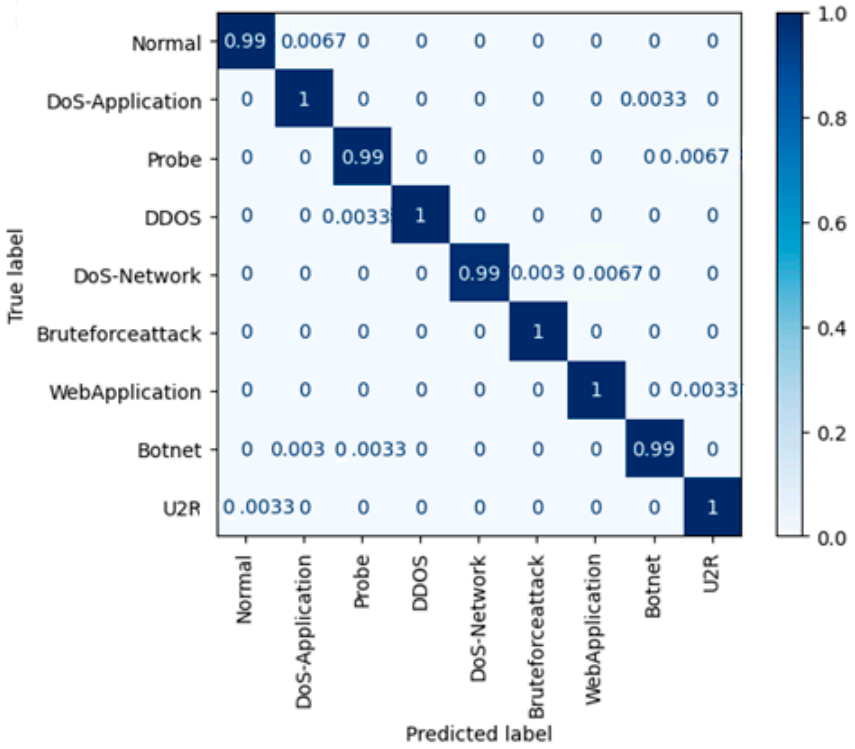


Figure 10. Confusion matrix for the InSDN dataset.

6. Conclusions and Future Work

In the context of smart cities, the importance of cybersecurity cannot be overstated. In these environments, the exponential growth of IoT devices exposes new vulnerabilities that may be exploited by malicious actors. In order to ensure the safety and secrecy of citizens' data and critical infrastructure, it becomes increasingly important. By utilizing deep learning techniques, particularly the SDEM, this study proposes an innovative approach to fortify the IoT infrastructure of smart cities, specifically SVN. This study demonstrates the effectiveness of the proposed SDEM in identifying cyberattacks within the IoT ecosystem of smart cities. With 99.8% accuracy for the ToN-IoT dataset and 99.6% accuracy for the InSDN dataset, this approach illustrates the potential of this approach for intrusion detection and cybersecurity. As evidenced by the SDEM, stacking offers promising results and represents a significant advancement.

The weakness can be addressed by combining deep learning with other security measures, including rule-based systems, anomaly heuristics, and physical security measures. It is also possible to mitigate privacy concerns when using deep learning for IoT security by focusing on data anonymization and privacy-preserving techniques. It is also crucial to monitor the performance of deep learning models in real-world IoT environments on a regular basis in order to maintain the effectiveness of security.

A future study in this area should investigate the scalability and robustness of the proposed model in real-world smart city environments. To ensure long-term security, it is imperative to take into account the dynamic nature of IoT systems, evolving cyberthreats, and the need for continuous adaptation. Additionally, research should focus on integrating the SDEM with operational smart city networks and assessing its performance under a variety of attack scenarios. In order to safeguard the digital backbone of modern urban life, advanced cybersecurity solutions will be vital to ensuring a safer, more efficient, and more resilient future for our cities, as smart cities continue to evolve.

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

Funding: This study is supported via funding from Prince Sattam bin Abdulaziz University (project number PSAU/2023/R/1444).

Data Availability Statement: The used dataset in this work (TON_IoT Datasets) is available Online. https://research.unsw.edu.au/projects/toniot-datasets (accessed on 3 April 2022).

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

References

1. Saba, T. Intrusion detection in smart city hospitals using ensemble classifiers. In Proceedings of the 13th International Conference on the Developments on eSystems Engineering (DeSE2020), Liverpool, UK, 14–17 December 2020.
2. Qureshi, T.N.; Khan, Z.A.; Javaid, N.; Aldegheishem, A.; Rasheed, M.B.; Alrajeh, N. Elephant herding robustness evolution algorithm with multi-clan co-evolution against cyber attacks for scale-free internet of things in smart cities. *IEEE Access* **2023**, *11*, 79056–79072. [CrossRef]
3. Prabakar, D.; Sundarrajan, M.; Manikandan, R.; Jhanjhi, N.Z.; Masud, M.; Alqhatani, A. Energy Analysis-Based Cyber Attack Detection by IoT with Artificial Intelligence in a Sustainable Smart City. *Sustainability* **2023**, *15*, 6031. [CrossRef]
4. Symantec. *Internet Security Threat Report*; Symantec: Tempe, AZ, USA, 2020; p. 1.
5. Costa, B.; Bachiega, J.; de Carvalho, L.R.; Araujo, A.P.F. Orchestration in Fog Computing: A Comprehensive Survey. *ACM Comput. Surv.* **2022**, *55*, 1–34. [CrossRef]
6. Chang, V.; Golightly, L.; Modesti, P.; Xu, Q.A.; Doan, L.M.T.; Hall, K.; Boddu, S.; Kobusińska, A. A survey on intrusion detection systems for fog and cloud computing. *Future Internet* **2022**, *14*, 89. [CrossRef]
7. Saba, T.; Rehman, A.; Sadad, T.; Kolivand, H.; Bahaj, S.A. Anomaly-based intrusion detection system for IoT networks through deep learning model. *Comput. Electr. Eng.* **2022**, *99*, 107810. [CrossRef]
8. Yan, Z.; Yang, G.; He, R.; Yang, H.; Ci, H.; Wang, R. Ship Trajectory Clustering Based on Trajectory Resampling and Enhanced BIRCH Algorithm. *J. Mar. Sci. Eng.* **2023**, *11*, 407. [CrossRef]
9. Liu, X.; Liu, Y.; Liu, A.; Yang, L.T. Defending ON–OFF attacks using light probing messages in smart sensors for industrial communication systems. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3801–3811. [CrossRef]
10. Guo, Y.; Wang, Y.; Khan, F.; Al-Atawi, A.A.; Abdulwahid, A.A.; Lee, Y.; Marapelli, B. Traffic Management in IoT Backbone Networks Using GNN and MAB with SDN Orchestration. *Sensors* **2023**, *23*, 7091. [CrossRef]
11. Sivaramakrishnan, R.; SenthilKumar, G. Workload Characterization in Embedded Systems Utilizing Hybrid Intelligent Gated Recurrent Unit and Extreme Learning Machines. *Int. J. Intell. Syst. Appl. Eng.* **2024**, *12*, 233–243.
12. Jagadeesan, J.; Kirupanithi, D.N. An Optimized Ensemble Support Vector Machine-Based Extreme Learning Model for Real-Time Big Data Analytics and Disaster Prediction. *Cogn. Comput.* **2023**, 1–23. [CrossRef]
13. Dixit, P.; Kohli, R.; Acevedo-Duque, A.; Gonzalez-Diaz, R.R.; Jhaveri, R.H. Comparing and Analyzing Applications of Intelligent Techniques in Cyberattack Detection. *Secur. Commun. Netw.* **2021**, *2021*, 5561816. [CrossRef]
14. Sriranjani, R.; Saleem, M.D.; Hemavathi, N.; Parvathy, A. Machine Learning Based Intrusion Detection Scheme to Detect Replay Attacks in Smart Grid. In Proceedings of the 2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 18–19 February 2023; pp. 1–5.
15. Deep, S.; Zheng, X.; Jolfaei, A.; Yu, D.; Ostovari, P.; Kashif Bashir, A. A survey of security and privacy issues in the Internet of Things from the layered context. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e3935. [CrossRef]
16. Patel, O.; Patel, H. To Use an Ethereum-Based Public Blockchain Network to Provide Confidentiality, Integrity, and Access Control to IoT-Based Medical Healthcare Data. *J. Pharm. Negat. Results* **2022**, *13*, 4400–4413.
17. Li, Z.; He, Y.; Yu, H.; Kang, J.; Li, X.; Xu, Z.; Niyato, D. Data heterogeneity-robust federated learning via group client selection in industrial IoT. *IEEE Internet Things J.* **2022**, *9*, 17844–17857. [CrossRef]
18. Rao, P.M.; Deebak, B.D. Security and privacy issues in smart cities/industries: Technologies, applications, and challenges. *J. Ambient Intell. Humaniz. Comput.* **2022**, *14*, 10517–10553. [CrossRef]
19. Anand, P.; Singh, Y.; Selwal, A.; Singh, P.K.; Felseghi, R.A.; Raboaca, M.S. IoVT: Internet of Vulnerable Things? Threat Architecture, Attack Surfaces, and Vulnerabilities in Internet of Things and Its Applications towards Smart Grids. *Energies* **2020**, *13*, 4813. [CrossRef]
20. Malhotra, P.; Singh, Y.; Anand, P.; Bangotra, D.K.; Singh, P.K.; Hong, W.-C. Internet of Things: Evolution, Concerns and Security Challenges. *Sensors* **2021**, *21*, 1809. [CrossRef]
21. Kumar, P.; Prabhat, G.P.; Tripathi, R.G. An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks. *Comput. Commun.* **2021**, *166*, 110–124. [CrossRef]

22. Said Elsayed, M.; Le-Khac, N.A.; Dev, S.; Jurcut, A.D. Network anomaly detection using LSTM based autoencoder. In Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks (2020), Alicante, Spain, 16–20 November 2020; pp. 37–45.
23. Elsayed, M.S.; Jahromi, H.Z.; Nazir, M.M.; Jurcut, A.D. The role of CNN for intrusion detection systems: An improved CNN learning approach for SDNs. In *Future Access Enablers for Ubiquitous and Intelligent Infrastructures, Proceedings of the 5th EAI International Conference, FABULOUS 2021, Virtual Event, 6–7 May 2021*; Springer: Cham, Switzerland, 2021; pp. 91–104.
24. Rondon, L.P.; Babun, L.; Aris, A.; Akkaya, K.; Uluagac, A.S. Survey on enterprise Internet-of-Things systems (E-IoT): A security perspective. *Ad Hoc Netw.* **2022**, *125*, 102728. [CrossRef]
25. Alajanbi, M.; Mohd Arfian Ismail, R.A.H.; Sulaiman, J. Intrusion Detection: A Review. *Mesopotamian J. Cybersecur.* **2021**, *2021*, 4.
26. Safara, F.; Souri, A.; Serrizadeh, M. Improved intrusion detection method for communication networks using association rule mining and artificial neural networks. *IET Commun.* **2020**, *14*, 1192–1197. [CrossRef]
27. Abdel-Basset, M.; Moustafa, N.; Hawash, H.; Ding, W. *Deep Learning Techniques for IoT Security and Privacy*; Springer: New York, NY, USA, 2022; Volume 997.
28. An, G.H.; Cho, T.H. Improving Sinkhole Attack Detection Rate through Knowledge-Based Specification Rule for a Sinkhole Attack Intrusion Detection Technique of IoT. *Int. J. Comput. Netw. Appl. IJCNA* **2022**, *9*, 169–178. [CrossRef]
29. Haider, W.; Moustafa, N.; Keshk, M.; Fernandez, A.; Choo, K.-K.-R.; Wahab, A. FGMCHADS: Fuzzy Gaussian mixture-based correntropy models for detecting zero-day attacks from linux systems. *Comput. Secur.* **2020**, *96*, 101906. [CrossRef]