Journal of Theoretical and Applied Information Technology
29th February 2024. Vol.102. No 4

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

INTRUSION DETECTION SYSTEM FOR (IOT) NETWORKS USING CONVOLUTIONAL NEURAL NETWORK (CNN) AND XGBOOST ALGORITHM

FIRAS H. ZAWAIDEH[1], GHAYTH AL-ASAD[2], GHAITH SWANEH[3], SARA BATAINAH[4],HUSSAIN BAKKAR[5]

[1-5]Department of Cyber Security, Faculty of Science and Information Technology, Irbid NationalUniversity, irbid, jordan

e-mail: [1]f.zawaideh@inu.edu.jo, [2]g.alasad@inu.edu.jo ,[3]g.sawaneh@inu.edu.jo , [4]s.bataineh@inu.edu.jo, [5]h.bakkar@inu.edu.jo

ABSTRACT

Recently, the Internet of Things systems has seen a significant growth, as it becomes a part of our daily activities. However, some related security issues of rapidly evolving IoT threats need to be solved regarding the traditional Intrusion Detection Systems (IDS). This work proposes a novel mechanism to overcome the challenges of the traditional IDS including the computational power, storage, and high rates of false alarms. Our mechanism uses Convolutional Neural Network (CNN) to extract features. The CNN architecture is effective at extracting significant features necessary for anomaly detection from the raw data collected by IoT devices. These features are fed to the XGBoost, which is superior at identifying intricate associations in the data, increasing the accuracy of intrusion detection. The combination of Deep learning and ensemble methods provides a robust solution to protect IoT environments against various attacks. An IoT *NetFlow*-based *dataset* was used, named NF-bot-IoT dataset, it contains four types of attacks within IoT network. The experiment is conducted using python programming language, and the Results indicate how effective this mechanism is, showing how it can detect a variety of attacks with high accuracy equals 98.76. This study helps to strengthen the IoT's security framework and increases its resistance to cyber threats.
Keywords: *CNN; IOT; Internet of things; security; deep learning; XGBoost algorithm*

1. INTRODUCTION

The development of various technology fields such as sensors, distributed services, embedded computing, automatic identification and tracking and wireless communications has raised the potential to incorporate smart objects into our daily lives. This integration among the physical devices and the internet defined as internet of things (IOT) [1]. This paradigm shifts the internet into a more connected environment. According to Marco research, the IoT may have 83 billion connections units by 2024, increasing from 35 billion connections in 2020 [2]. Also, it is anticipated that by 2024, the industrial IoT (IIoT) sector—which includes manufacturing, agriculture, and retail will account for more than 70% of all IoT connections, with a 180% increase in IIoT units during the following four years [2].

IoT systems may provide significant advantages fordifferent application domains, includingtransportation, public safety, energy, industrial processes, home automation, healthcare, and environmental monitoring [3]. However, IoT systems are vulnerable to different cyber security attacks. Some of the attacks may results in big consequences to the countries like attacks against the transportation, or power plants. Additionally, Homeappliances might be a target, threatening the privacyand security of families. This paper [4] performed a test for three common home appliances including Phillips Hue light-bulb, the Nest smoke-alarm and the Belkin WeMo power switch. The results showed the vulnerability of these devices to the privacy and security. However, traditional security protection measurements may not work efficiently on IOT devices, due to the low computational power, and the high number of devices.

Intrusion Detection System (IDS), is a system security technique that entails identifying the malicious activities over the network or system. Its main function is to detect the abnormal behaviors that may contains some security intrusion. There are two types of IDS, network based Intrusion Detection System (NIDS) and host based

Intrusion Detection System (HIDS) [5]. The NIDS is responsible to investigate data packets and payloads to detect the abnormal behavior aiming to monitor the network traffic in real-time and generate responses. While HIDS responsible to perform the monitoring of activities occurring on individual host systems like servers, by investigating the system logs and user behaviors to identify abnormal behaviors. It plays an essential role in assisting the organizations to detect and respond to different attacks, improving the system cyber security in efficient time. However, due to the resources complexity and limitations, IDS are not secure enough for IoT systems. Furthermore, most of the intrusion detection systems require a lot of computing power and storage capacity. On the other hand, they may suffer from high rates of false alarms that can lead to ignore harmful attacks.

Hence, there is an extensive need to develop a robust and modern IDS solution that can protect IoT systems from attacks with less computational power and storage, and can overcome all these challenges of the traditional IDS.

Recently, deploying Artificial Intelligence (AI) for intrusion detection, becomes one of the innovative and efficient solutions [7]. Artificial Intelligence (AI) is a technology that permits the computer device to imitate human behavior such as learning and problem solving, as it finds innovative ways to analyze data. ML is a subset of AI that permits a machine to learn from data without having to be programmed explicitly [8] to discover patterns from data and predict the output. These models automatically learn from historical data [9]. AI and ML models play a vital role to solve security issues and intrusion detection tasks particularly, due to its capabilities to handle such problems and build intelligence activities. Such as, support vector machines (SVM) [10], genetic algorithms (GA), and Bayesian networks [11]. While the deep learning (DL) is a subset of ML that employ the neural networks to perform different operations, Neural networks mimic how the human brain functions to find the complex associations in the dataset. It uses multiple layers in order to extract significant features progressively from the input and then map between input and output [12]. DL-based intrusion detection systems for IoT networks is still under-researched. Hence, this research use NN and add a value to the existing literature.

According to the literature, the attacks detection task considered a classification problem, because the main target in this scenario is to identify whether the packet is normal packet or attack packet. Therefore, the IDS model can be developed using ML algorithms [13]. This work performs the detects different attacks aiming to identify the intrusions from IOT networks exploiting the power of deep learning in feature extraction and the ability of machine learning algorithm to perform accurate classification of the presence of attack and overcome all of the traditional IDS issues.

2. RELATED WORK

Several innovative techniques have been developed to solve the challenges of security in IoT networks. The literature has focused on introducing deep neural networks to detect intrusion in these environments effectively. This paper [14] provides intrusion detection method using various deep learning algorithms like Convolutional Neural Network (CNN), Gated Recurrent units (GRUs) and long short-term memory (LSTM), they compare among all of these algorithms and identify the best one of them to detect intrusion in IoT. They used standard dataset for testing named Bot-IoT. The findings of their study showed that the LSTM was the best algorithm with accuracy of 99.8% and recall of 100% and 99.7% of precision. And they outperformed the accuracy of the methods in the literature, thus it can be used for intrusion detection in IoT networks.

Also, the authors of this study [15], investigated a new method for intrusion detection by investigating the capabilities of two types of NN which are self- normalizing neural network (SNN) and feedforward neural network (FNN). They used Bot-IoT dataset to test the algorithms, the results demonstrated that the FNN achieved the highest performance among two, with 95.1% accuracy score, and the average of F1- score, precision and recall was 0.95%. However, they performed the feature normalization to adversarial attacks to test the adversarial robustness, they observed that, it is negatively affected the performance of both networks, by lowering the accuracy score to the half, which is inappropriate performance for real-life applications. But still SNN is considered more resilient by 9% accuracy than FNN.

This work [16] used two datasets the first one is KDD CUP 1999 dataset to detect four

types of Denial-of-Service Attacks such as remote to local (R2L), DoS, user to root (U2R), and probing. And the second dataset named CSE-CIC-IDS2018, this dataset contains more advanced DoS attacks like DoS-SlowHTTPTest, DoS-Hulk, DoS-Slowloris, DoS-GoldenEye, DDoS-HOIC, DDoS-LOIC-HTTP. They used Convolutional Neural Network (CNN) and compare it with Recurrent Neural Network (RNN) aiming to detect different attacks categories. They generated two kinds of intrusion image, grayscale and RGB, as well as they performed binary and multiclass for classification for different scenarios. The findings showed that, in both binary and multiclassification, the RGB images attained higher performance results. CNN model achieved 99% accuracy on the first dataset in both binary and multiclass classifications. While the RNN achieved 99% and 93% consequently for binary and multi class. Moreover, when they applied their model on the second dataset, the CNN achieved an average accuracy of 91.5%, and the RNN achieved 65% on average. In other words, when compared to the RNN model, the CNN model was more capable to identify particular DoS attacks that have the samefeatures.

This paper [17] perform four feature extraction techniques to detect intrusions in IoT devices, including employing DL models like DenseNet andVgg16, the dataset used was IEEE Dataport imagedataset. And some image filters like auto-colorcorrelogram and FcTH filters to map image featuresinto feature space. For classification phase, they useboth individuals ML models like (KNN) andRandom Forest, and stacked ML models like (KNNwith SMO). The results demonstrated that theirapproach is superior in extracting attacks fromimages, with 98.3 accuracy score for Vgg16 modelas features extractor and stacking KNN with SMO. This paper [22] used XGBoost to detect the intrusionusing two imbalanced IoT datasets which are X-IIoTDS and TON_IoT, the findings of the studyshowed that the proposed algorithm was superior indetecting different attacks, with f1 score equals99.9% and 99.87% on the two datasets respectively,X-IIoTDS and TON_IoT. Moreover, their resultsdemonstrated that the model solved the imbalanceissue and outperformed the existing methods in the literature.

This work [23] proposed ML algorithm to detect intrusion, based on Decision tree and random forest(RF), as well as these models don't demand high computing power to train. They used NF-BoT-IoT- v2, and NF-ToN-IoT-v2, and IoTDS20 datasets for experiments. The SHapley additive exPlanations (SHAP) is performed to the eXplainable AI (XAI) methodology to interpret the classification results from DT and RF. This can help decision makers to take well informed judgments based on the obtained results. They achieved 100% accuracy and F1 score, and outperformed the results in the literature.

3. METHODOLOGY

This section provides the methodology of this work including the dataset used, preprocessing, feature extraction and a description for all of the methods used to implement the proposed mechanism. The purpose of the modeling is to recognize the malicious traffic from the normal traffic. The firstsubsection displays the dataset description.

3.1 Dataset

The dataset used in this research is the open-sourcedNF-bot-IoT-v2 dataset, which was provided by [18],this dataset is an expanded version to the NF-BoT-IoT dataset. It contains extended 43 Net Flow characteristics, which was extracted from Pcap files. Each flow in the data is classified with the respective attack type. The number of flows is 37,763,497. 0.36% of them is benign flows. The rest of flows categorized into 4 attacks types; the next table shows the distribution of the classes in this dataset:

Table1: Dataset Class Distribution

Class	Count
Benign	135037
Reconnaissance	2620999
DDos	18331847
Dos	16673183
Information Theft	2431

3.2 Methods Used

This section explains the methods used in this work including the convolutional neural network and XGBoost algorithms, how it works and what is the essential components of both of them. First the CNNis explained.

3.2.1 CNN

Convolutional Neural Network (CNN) is one deep neural network type that mimic the human brain

function and extract relevant features from the row data. It contains several kinds of layers such as:

- **Convolution layers**

This type of layers is the building block of CNN, which is responsible of extracting hidden patterns from the data. These layers apply filters to extract features (low-level and high-level features), which are simple and complex features. These filters move on the input data and perform some mathematical calculations to produce something called feature maps, each filter produce one feature map showing the pattern of the input. The values in the feature map represent the strength or intensity of the observed feature at various input positions. There are Conv1D for sensor data for example, and Conv2d for images(represents pixels and channels) [19]. In this study, CONV1D was applied to identify the intrusion in theIoT flows.

- **Pooling Layers**

These types of layers are performed over the feature maps that were produced from the convolutional filters to down sampling them, this operation ensure

reducing the dimensionality of the extracted features and maintaining the important ones. In AvgPooling the window is slide over the feature map and take theavg of all values to represent the feature map. This process reduces the computational complexity and maintaining the significant features.

- ***Fully- Connected Layers***

Fully connected layers enable the network to learn intricate feature combinations, enabling it to identify complex associations in the data. In Fully connectedlayers, all of the neurons in the previous layers and next layers are connected.

3.2.2 XGBoost

Extreme Gradient Boosting, often known as XGBoost, is a powerful Gradient Boosting Decision Tree (GBDT) method that effectively uses both software and hardware optimization approaches to deliver higher results while using memory more efficiently. XGBoost can prevents the overfitting, as well as it supports both LASSO (L1) and Ridge (L2) regularization [20]. The XGBoost algorithm computes a set of features at each iteration that are significant for to classify the target. These features are utilized to divide the data into smaller groups according to their values. The subgroups are divided. Until the model can no longer improve, or until a predetermined stopping criterion is satisfied. The main purpose of this algorithm is to learn a function that can map input to the output. XGBoost add DT iteratively aiming to enhance its performance, these trees are built using Boosting process, to make the next layer learn from the error of the previous tree (misclassified).

3.3 Workflow of the proposed methodology

This section includes the proposed methodology of this work, including the steps implemented to perform the classification of the IoT flows into 5 categories, including Benign, Reconnaissance, Ddos, Dos and information theft. The following diagram shows the flowchart of the proposed methodology:

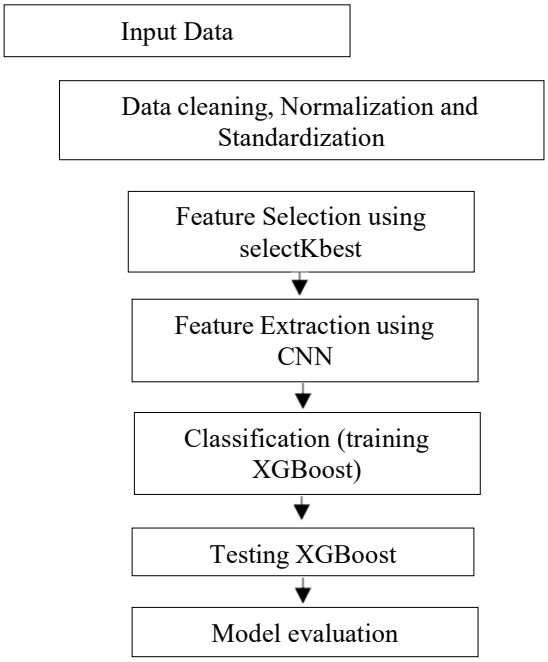

Figure1: The Flowchart Of The Proposed Methodology

Our mechanism as depicted in the above figure combines two algorithms to provide more robust results, it employs the Convolutional Neural network (CNN) as a feature extractor and XGboost as a classifier to detect the intrusion in IoT devices. Firstof all the numerical Bot- IoT dataset is inputted, then some of the preprocessing steps is applied in order to prepare the data to enter the model, these steps including filling missing values by 0 or by numerical value, mapping categorical features, removing duplicate samples, standardization to makes sure that all of the data are on the same scale and have a standard normal distribution

value between 0 and 1,also normalizing the dataset which is a transformation step of the values between 0 and 1, aiming to avoid negative values. The next step is the feature selection to select the most important features, by using select Kbest function. This function select the best features based on the k highest score by tuning the 'score_func' parameter. Then the train test split is performed in order to divide the dataset into a particular ratio (80:20 or 90:10) for both training and testing. After that the CNN model is trained to capture the complex patterns from the preprocessed data, these features are then sent to the XGboost, which uses a group of decision trees to accurately classify the extracted and significant features into 5 classes. The final stage of our mechanism is to evaluate our hybrid model that combines two methods using some evaluation metrics such as, accuracy, precision and recall.

Ensuring that the proposed algorithm can be used inthe real-life applications.
The following figure shows the architecture of the proposed CNN model that had been used for features extraction, it is designed to extract the complex associations and the hidden patterns from the Bot-IoT data after preprocessing.

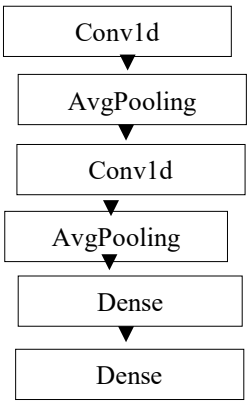

Figure2: Architecture Of CNN

As seen in the above figure, our CNN model consists a set of layers, starting by conv1D layer which responsible of extracting features from data, it convolves with the input using a set of filters to detect the low-level features, following by average pooling to reduce the learned features dimensionality. The second con1d layer also applies more filters to extract the high-level features and refine the previous extracted ones, additionally a maxpooling layer is added to perform down sampling of the features, reduce the dimensionality and the complexity of computation. These extracted features are flattened in dense layer (fully connected layer), these layers permit the network to learn the intricate combinations of features and extract the complex associations from the data. by the classifiers. Table2 below shows a confusion matrix of binary classification (two classes).

Table2: Confusion Matrix

	Predictive Positive	**Predictive Negative**
Actual Positive	True Positive	False Negative
Actual Negative	False Positive	True Negative

The True Positive value represents to the quantity ofthe positive samples and classified as positive, while the False Positive refers to the number of positive selections that are wrongly classified. Also, FN refers to the number of negative samples that are classified as Positive. However, True Negative is the number of Negative samples classified as negative [21]. Based on the CM Table, we can measure the performance of the algorithms and evaluate them through various empirical measures, as the follows [21]:

- **Accuracy**

This measure describes how the classifier performed at all classes of the data. It represents the correct predictions ratio that a trained classifier achieves. For binary classification, it can be calculated through the following equation:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \dots (1)$$

- **Precision**

This metric can be used to determine what proportion of retrieved examples are relevant to the overall number of retrieved examples. Precision is calculated as follows:

The last layer is responsible to perform the classification. The model is then trained and saved, after that it is loaded

without the last layer, aiming to use the learned high- level features as input to the XGBoost algorithm. The compilation of this model includes the Adam optimizer and the Sparse Categorical Cross Entropyas loss function.

3.4 Evaluation Metrics

To evaluate our proposed method, the Confusion matrix (CM) was used: to evaluate the performance of the classifier. In this matrix, there are significant information about the predictions that are expected

the XGboost after the CNN feature extraction process, in terms of confusion matrix information and visual presentation of the results. Moreover, it shows a comparison with the related work from the literature [15, 16, and 17].

4.1 Hardware and Software Requirements

There are a set of requirements (software and hardware).

- **Software Requirements**

There are particular set of software components essential to use the software successfully such as:1-Python Interpreter: A Python interpreter is required to execute the system. We used python 3.9.

2- Anaconda: Anaconda is a comprehensive environment and the birthplace of for data science and python. It provides a set of libraries, tools, and packages that help in setting up the system's dependencies.

3-Visual Code: It is a powerful integrated development environment that provide use friendly interface for development and debugging. Also, it supports python and many other programming languages.

4- Libraries:

- Numpy: it is a main Python package for numerical computations. It provides powerful objects like arrays. As it allows for efficient manipulation of arrays and matrices, facilitating various mathematical tasks.
- Pandas: it's also an open-source library for data analysis and manipulation, as it provides powerful data structures that used to analyze data such as dataframe and series.

$$Precision = \frac{TP}{TP+TFN} \quad \ldots. (2)$$

- **Recall**

This measure called recall or sensitivity, and it refers to the number of positive examples that are correctly predicted made out of the whole number of positiveexamples.

$$Recall = \frac{TP}{((TP+FN))} \quad \ldots (3)$$

4- Results and discussions

This chapter displays the software and hardware requirement that had been used to perform the implementation of this work using python language, the results obtained, the training results of CNN, and

- Keras: it's a deep learning library that can be used to train and test the deep neural network algorithms. And fine-tuning different hyper-parameters.

• XGBoost Library: This library responsible of building the XGBoost algorithm easily.

- **Hardware Requirements**

The system's efficient performance depends on suitable hardware components. During the software execution, the following hardware specifications were used:

1- Memory (RAM): 16 RAM is used to handledata efficiently, and to reduce the possibility of slowdowns.

2- Processor (CPU): we utilized an Intel Core i7 processor, boasting a clock speed of 3000 MHz to ensure that complex operations are completedspeedily without delays.

3- Storage (SSD): we add a 512 GB (SSD) toexpand the system's capabilities.

4.2 CNN Model training Results

This section displays the CNN model training resultsthrough its layers over a set of epochs.

- ***Loss analysis of CNN***

The loss function measures the difference among the predicted and the actual results during the CNNtraining, it is an indicator of how well the model learning to extract relevant features from the flows IoT data. A lower loss value signifies that the model is effectively capturing the underlying patterns in thedata.

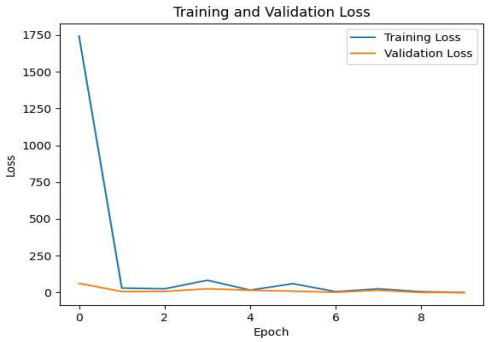

Figure3: CNN loss

We analyzed the loss value progression throughout the training and validation phases. By monitoring the loss trend, we gained insights into the model's abilityto optimize its parameters and minimize errors. The decreasing trend indicates that the CNN model is successfully learning and refining its feature extraction capabilities. As depicted in the abovefigure. It illustrates that the Loss of the model was very large at the first epoch, which indicates that the initial prediction performance contains errors. Then the loss is dropped in the next layer, improving its performance and reducing the dispersion among thepredicted and the actual values. As training continues, the loss decreases, indicating continuous enhancement in the model's accuracy. The CNN model captures the complex trends from the input IoT data and improves its learned features. The finalloss showcases the high capability of the model in feature extraction from data, as it reduces the most significant and useful features of the input into its hidden state, offering good data representation.

- ***Accuracy analysis of CNN***

This metric is very significant, that evaluate the CNN ability to classify the attacks correctly based on the extracted features. A higher accuracy indicates that the GRU model is effective in capturing discriminative features that distinguish between benign and different attacks samples. See the following figure that shows the model accuracy overa set of epochs.

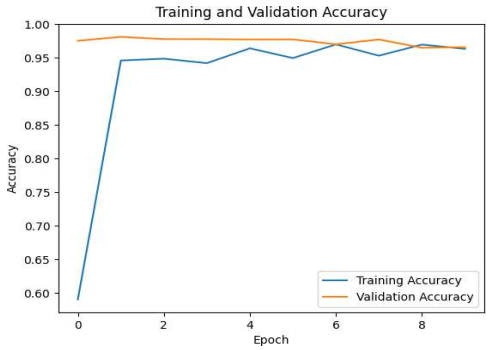

Figure4: CNN accuracy

The accuracy in figure4 illustrates the development of accuracy throughout the training and validation over 10 epochs, at the beginning, it was below 60% which is not reliable, however as the training goes, the accuracy gets better, by the final epoch, the validation accuracy reaches 98.6, showing a significant improvement in the ability of the model to classify the inputs accurately. The accuracy graphpresents the model's ability to learn as it iteratively improves its predictions by capturing relevant patterns and features from the training data. This increase in accuracy reflects the model's effectiveness in understanding the underlying relationships within the input data, leading to improved performance over time.

4.3 XGBoost Confusion Matrix Results

- XGBoost Confusion Matrix (CM)

Aiming to evaluate the results of the XGBoost algorithm for multiclass classifications, we employed the confusion matrix. As it was explained in section3. We found 3 metrics from the CM information, including, accuracy, precision and recall. The accuracy describes how the

classifier performed at all classes of the data. It represents thecorrect predictions ratio that a trained classifier achieves. Precision rate represents the true positive (TP) predictions rate over the total predicted positive instances for a specific class, while the recall or True positive rate refers to the actual positive samples ratethat were correctly recognized

```
[[   74     0   121    45     0]
 [    0 29040   160     2     0]
 [   10   143 25966   293     0]
 [    5     0   270  3866     0]
 [    0     0     1     0     4]]
```

Figure5: CM

As shown in the previous figure, which illustrates theconfusion matrix for multi class in IoT data (DoS, Ddos, Reconnaissance, Benign, and Theft). For class1 (DoS), our mechanism identified 74 examplesas DoS, however it missed 121 example that incorrectly classified as other classes. And no examples incorrectly classified as DoS.
While the model performed efficiently on class2 in the data which is (Ddos), it was correctly identified 29040 examples and only it misclassified 162 as DdoS. And miss 2 examples that were classified to other classes. Additionally, the model was also efficient to detect Class3 in the data which is Reconnaissance, by recognizing 25966 instances correctly, 293 missed and identified as other classes. And 143 classified as Reconnaissance but it wasn't actually.
Benign class, which hold the normal examples, there were 3866 benign examples identified correctly, missed only 5 examples as benign. However, there were 270 ones from other classes inaccurately identified as benign. Finally, theft class had been identified correctly by the proposed mechanism in 4 examples, misclassified 1 as theft but it wasn't, and didn't miss and theft examples.
After the analysis of CM values, we can compute different measures to test the model performance at all. Aiming to understand how it performs and if it's suitable to use in the real-life applications. For example, the accuracy of the proposed mechanism was 98.67 which indicates that XGboost was powerful in identifying different attacks types.
See the next figure, which shows the recall andprecision for all classes:

	precision	recall	f1-score	support
0	0.96	0.30	0.46	240
1	1.00	0.99	0.99	29202
2	0.98	0.98	0.98	26412
3	0.92	0.93	0.93	4141
4	1.00	0.80	0.89	5

Figure6: Recall And Precision For All Classes

The figure shows that recall for class 2 and 3 is nearly perfect, while in the others it seems there is some classification error. But overall, the model is performed well in all of them, making the model reliable and can be used in the real-life applications.

- Heatmap of the Confusion Matrix (CM)

The heatmap represents the confusion matrix for the model, providing a visual summary of the model's performance across different classes. Each cell in theheatmap correspond to a specific value of CM, such as TP, TN, FP, or FN for a particular class. The colors in the heatmap indicate the values of the metrics, with darker shades representing higher values. When the model generates good results and have some stability, then the corresponding heatmap will appear a diagonal of dark colors, as well as these dark areas present the correctly classified ratios, See the heatmap of the proposed approach in figure7:

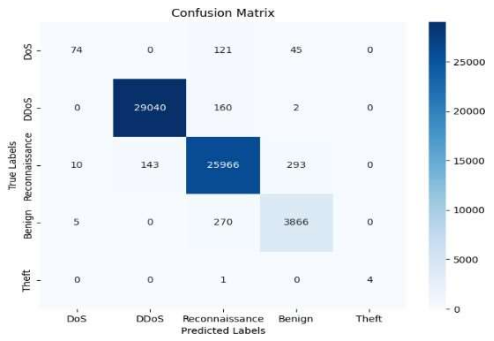

Figure7: Heatmap of the obtained results

From analyzing the heatmap, we can observe that the XGBoost achieved good results values for most classes. Classes 2 and 3 stand out with near-perfect scores, indicating that the model successfully predicted instances belonging to these classes with a high level of accuracy. However, some classes exhibited some lower performance metrics.

Overall, the XGBoost demonstrated good performance in term of accuracy, recall and precision.
By observing the results of class2 in the above heatmap. The model was correctly identified 29040 examples and only it misclassified 160 as DdoS. And miss 2 examples that were classified to other classes. In addition, for chlass3 the model was also efficient,as it was recognized 25966 instances correctly, 293 missed and identified as other classes. And 143 classified as Reconnaissance but it wasn't actually. Noting that the explanation of this heatmap taken from the Confusion matrix values itself but in visual form. Overall, the model was good enough to classify different types of intrusions as depicted fromthe illustration.

4.4 Comparison with related work

This section provides a comparison of our work withthe literature, see the next table:

Table3: Comparisons Of Our Approach With TheLiterature

Ref	Technique	Result	Dataset	Datase ttype
[15]	feedforward NN	95.1%	Bot-IoT dataset	numerical
[17]	Vgg16 + KNN withSMO	98.3%	IEEE Dataport image datase t	imaginary
[16]	CNN	99% 91.5	KDD CUP 1999 CSE-CIC-IDS201 8	imaginary
[22]	(XGBoost)	99.9% 99.87 %	X-IIoTDS TON_Io T datasets	numerical
Our approac h	CNN + XGBoost	98.867	Bot-IoT dataset	numerical

4.5 Conclusion

This work proposed a new mechanism, and use NF-Bot-IoT data in order to detect various attacks. Ourmain function is to detect the abnormal behaviorsthat may contains some security intrusion. First ofall, the data is preprocessed and then the best 15 features were selected based on selectKbest method,here are some of these features:
['L4_DST_PORT','PROTOCOL',
'L7_PROTO',
'TCP_FLAGS',
'CLIENT_TCP_FLAGS',
'SERVER_TCP_FLAGS',
'FLOW_DURATION_MILLISECONDS',
'DURATION_IN',
'ICMP_TYPE',
'ICMP_IPV4_TYPE'].
After that, these features are entered to the CNN model for feature extraction step aiming to extract the hidden patterns and complex associations from the data. The CNN model was built on Keras libaraywith 6 layers (2 conv1D, 2 avgPooling, and 2 dense layers). The CNN model was very efficient, its performance was proved by the loss and accuracy scores on both training and validation. On the other hand, its total parameters was 124,096, suitable, avoided the over fitting, and increased the model generalization.
Finally, the extracted features from CNN are fed as input to the XGBoost algorithm, aiming to combine the high capability of Neural Network in feature extraction through its layers, and the power of ensemble methods in performing the classification efficiently. The results of this study showed the robustness of the proposed mechanism. Evident by the recall and precision values for all classes.

REFERENCES:

[1]. Zarpelão, B. B., Miani, R. S.,Kawakani, C. T., & de Alvarenga, S.
C. A survey of intrusion detection in Internet of Things. *Journal of Network and Computer Applications*, (2017) *84*, 25-37.
[2]. Centenaro, M., Costa, C. E., Granelli, F., Sacchi, C., & Vangelista, L. A survey on technologies, standards and open challenges in satellite IoT. *IEEE Communications Surveys & Tutorials*,2021, *23*(3), 1693-1720.
[3]. Borgia, E., The Internet of Things vision: Key features, applications and open issues. *ComputerCommunications*, 2014, *54*, 1-31.
[4]. Notra, S., Siddiqi, M., Gharakheili, H. H., Sivaraman, V., & Boreli, R. An experimental study of security and privacy risks with emerging household appliances. In *2014 IEEE conference on communications and network*

security, 2014, (pp. 79-84). IEEE.

[5]. Ashiku, L., & Dagli, C. Network intrusion detection system using deep learning. *Procedia Computer Science*, 2021, *185*, 239-247.

[6]. Michie, D., Spiegelhalter, D. J., & Taylor, C. C. Machine learning, neural and statistical classification, (1994).

[7]. Alohali, M. A., Al-Wesabi, F. N., Hilal, A. M., Goel, S., Gupta, D., & Khanna, A. Artificial intelligence enabled intrusion detection systems for cognitive cyber-physical systems in industry 4.0 environment. *Cognitive Neurodynamics*, 2022, *16*(5), 1045-1057.

[8]. Shan, T., Tay, F. R., & Gu, L. (). Application of artificial intelligence in dentistry. *Journal of dental research*, 2021, *100*(3), 232-244.

[9]. Musa, U. S., Chhabra, M., Ali, A., & Kaur, M. (2020, September). Intrusion detection system using machine learning techniques: A review. In *2020 international conference on smart electronics and communication (ICOSEC)* (pp. 149-155). IEEE.

[10]. Kumari, V. V., & Varma, P. R. K. (2017, February). A semi-supervised intrusion detection system using active learning SVM and fuzzy c-means clustering. In *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I- SMAC)*, 2017, (pp. 481-485). IEEE.

[11]. Alhakami, W., ALharbi, A., Bourouis, S., Alroobaea, R., & Bouguila, N. Network anomaly intrusion detection using a nonparametric Bayesian approach and feature selection. *IEEE access*, 2019, *7*, 52181-52190.

[12]. Kim, K., & Aminanto, M. E. Deep learning in intrusion detection perspective: Overview and further challenges. In *2017 International Workshop on Big Data and Information Security (IWBIS)* (pp. 5- 10). IEEE., 2017

[13]. Almseidin, M., Alzubi, M., Kovacs, S., & Alkasassbeh, M. Evaluation of machine learning algorithms for intrusion detection system. In *2017 IEEE 15th international symposium on intelligent systems and informatics (SISY)* 2017, (pp. 000277-000282). IEEE.

[14]. Banaamah, A. M., & Ahmad, I. Intrusion Detection in IoT Using Deep Learning. *Sensors*, 2022, *22*(21),8417.

[15]. Ibitoye, O., Shafiq, O., & Matrawy, A. Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks. In *2019 IEEE global communications conference (GLOBECOM), 2019,* (pp. 1-6). IEEE.

[16]. Kim, J., Kim, J., Kim, H., Shim, M., &Choi, E. (2020). CNN-based network intrusion detection against denial-of- service attacks. *Electronics*, *9*(6), 916.

[17]. Musleh, D., Alotaibi, M., Alhaidari, F., Rahman, A., & Mohammad, R. M. (2023). Intrusion Detection System Using Feature Extraction with Machine Learning Algorithms in IoT. *Journal of Sensor and Actuator Networks*, *12*(2), 29.

[18]. https://www.kaggle.com/datasets/dhoogla/nfbotiotv2

[19]. Mekruksavanich, S., Jitpattanakul, A., & Hnoohom, N. Negative emotion recognition using deep learning for thai language. In *2020 joint international conference on digital arts, media and technology with ECTI northern section conference on electrical, electronics, computer and telecommunications engineering (ECTI DAMT & NCON),* 2020, (pp. 71-74). IEEE.

[20]. Ayus, I., Natarajan, N., & Gupta, D. Comparison of machine learning and deep learning techniques for the prediction of air pollution: a case study from China. *Asian Journal of Atmospheric Environment*, 2023, *17*(1), 4.

[21]. Sokolova, M., Japkowicz, N., & Szpakowicz, S. Beyond accuracy, F- score and ROC: a family of discriminant measures forperformance evaluation. In *Australasian joint conference on artificial intelligence*, 2006, (pp. 1015- 1021). Springer, Berlin, Heidelberg.

[22]. Le, T. T. H., Oktian, Y. E., & Kim, H. XGBoost for imbalanced multiclass classification-based industrial internet of things intrusion detection systems. *Sustainability*, 2022, *14*(14),8707.

[23]. Le, T. T. H., Kim, H., Kang, H., & Kim, H. Classification and explanation for intrusion detection system based on ensemble trees and SHAP method. *Sensors*, 2022, *22*(3), 1154.