International Journal of Artificial Intelligence, Data Science, and Machine Learning
Grace Horizon Publication | Volume 3, Issue 3, 73-81, 2022
ISSN: 3050-9262 | https://doi.org/10.63282/3050-9262.IJAIDSML-V3I3P108

Original Article

Enhancing Cybersecurity in Industrial Through AI-Based Traffic Monitoring IoT Networks and Classification

Anand Polamarasetti[1], Rahul Vadisetty[2], Srikanth Reddy Vangala[3], Varun Bodepudi[4], Srinivasa Rao Maka[5], Gangadhar Sadaram[6], Manikanth Sakuru[7], Laxmana Murthy Karaka[8]
[1]MCA, Andhra University.
[2]Wayne State University, Master of Science.
[3]University of Bridgeport, Computer Science Dept.
[4]Applab Systems Inc, Computer Programmer.
[5]North Star Group Inc, Software Engineer.
[6]Bank of America, Sr DevOps/ Open Shift Admin Engineer.
[7]JP Morgan Chase, Lead Software Engineer.
[8]Microsoft, Senior Support Engineer.

Abstract *- Industrial Internet of Things (IIoT) networks are very important to modern industry because they make it possible to watch, manage, and improve processes as they happen. As IoT systems are connected to each other, they can be attacked online, have system failures, and have trouble talking to each other. This paper proposes an AI-based traffic monitoring and classification framework to enhance cybersecurity in IIoT environments. Utilizing Bot-IoT datasets, the proposed system applies a Long Short-Term Memory (LSTM) model for real-time intrusion detection and threat classification. The methodology includes comprehensive data preprocessing techniques such as data cleaning, timestamp handling, one-hot encoding, feature selection, and normalization to ensure model robustness and accuracy. The LSTM model does better in experiments than common predictors like Random Forest and Naïve Bayes, with an F1-score of 99.87%, an accuracy of 99.74%, a precision of 99.99%, a recall of 99.75%, and an actual accuracy of 99.74%. These outcomes validate the effectiveness of deep learning in identifying and mitigating cyber threats in IIoT networks. The proposed model lays the groundwork for integrating intelligent cybersecurity mechanisms into future IIoT infrastructures to improve resilience and operational safety.*

Keywords *- Industrial IoT (IIoT), Cybersecurity, Intrusion Detection, LSTM, Deep Learning, Traffic Classification, Bot-IoT Dataset, Machine Learning, Anomaly Detection.*

1. Introduction

The IIoT has revolutionized industrial operations by integrating traditional industrial automation systems with advanced IoT technologies. This convergence has enabled seamless connectivity between devices such as sensors, robots, mixing tanks, and control systems across sectors including energy, healthcare, and automotive [1]. As IIoT evolves, it facilitates real-time data collection, advanced analytics, and intelligent decision-making, leading to improved product quality, enhanced production efficiency, and reduced operational costs [2][3]. One critical component of this technological transformation is traffic monitoring, which involves collecting and analyzing data on system activity and network usage. Originally prominent in Intelligent Transportation Systems (ITS), traffic monitoring now plays a pivotal role in industrial environments as well. In IIoT networks, traffic monitoring helps detect anomalies, optimize system performance, and improve safety by analyzing parameters such as device activity, data transmission rates, and network congestion. There are three main ways to deal with security problems in any network: prevention, monitoring, and mitigation [4]. All three of these steps will be needed for IoT network security options to work.

This study is mostly about Intrusion Detection Systems (IDS), and they look at DL-based IDS for finding and sorting network traffic in an IoT setting. To address these challenges, this study proposes an AI-based traffic monitoring and classification approach to enhance IIoT cybersecurity. By employing ML algorithms like RF, SVMs, and CNN, the proposed system aims to detect anomalies in real-time, classify malicious traffic, and provide adaptive responses to potential threats [5]. This AI-driven strategy offers a proactive cybersecurity framework capable of evolving with emerging risks [6]. Anomaly identification is also very important for finding behavior changes that aren't normal in IIoT networks. In sectors such as healthcare, where errors can be life-threatening, robust anomaly detection mechanisms are indispensable [7][8]. ML and DL techniques have shown great promise in enhancing the accuracy and efficiency of such systems.

1.1. Motivation and Contribution

The main purpose of this study is to safeguard IIoT networks by developing a system for AI-based traffic tracking and sorting. By leveraging DL techniques, particularly LSTM networks, the study seeks to accurately detect and classify various cyber threats in real-time, thus improving the resilience, reliability, and security of IIoT infrastructures against sophisticated intrusion attempts. There are some key contributions as follows:

- The Machine and DL-based approach for the classification of cybersecurity using the Bot-IoT dataset.
- Presented a structured preprocessing pipeline that includes feature normalization, feature selection for important traits, and one-hot encoding for categorical data to make the model work better.
- The study employs and compares multiple ML models (LSTM, RF, and NB) for intrusion detection using the Bot-IoT datasets.
- F1-score, accuracy, precision, and memory were used to get a big picture of the model's success.

1.2. Structure of the paper

The study is organized as follows In Section II, the existing literature on cybersecurity and IIoT networks is reviewed In Section III methodology utilized to compile the data for this study. Section IV provides the results and analysis of cybersecurity of traffic monitoring and classification. At last Section V provide the conclusion provides the conclusion.

2. Literature Review

This section discusses the Literature review on AI-Driven Traffic Monitoring for Enhanced Cybersecurity in Industrial IoT Networks. Also, Table I provides a summary of the literature reviews discussed below:

Dawoud, Shahristani and Raun (2019) Thinking about network anomaly detection again to see how DL could be used to find network risks. DL systems that learn without being watched are what we're studying. Unsupervised DL methods are used in the study to come up with a semi-supervised detection framework. Through autoencoders, a type of non-probabilistic method, the study looks at the pros and cons of using DL to find bugs. For finding anomalies, they give an in-depth study for AE. The USDL would improve identification, and their tests show that it would work over 99% of the time [9].

Nagisetty and Gupta (2019) use the Keras DL Library to show a way to find harmful activity in IoT Backbone Networks. Four types of DL models are used in the suggested framework to predict malicious attacks. These are the MLP, CNN, DNN, and Autoencoder. This test checks how well the base is built using two well-known datasets: UNSW-NB15 and NSL-KDD99. With the precision, RMSE, and F1-score, they can look at the numbers [10].

Zolanvari, Teixeira, and Jain (2018) Algorithms for ML have been shown to work well for protecting IT system platforms. But because IIoT networks are not the same as regular IT networks, performance needs to be looked at in a different way. Different things need to be thought about because IIoT systems are vulnerable and need to be secure. As part of this paper, they look at why ML needs to be a part of the IIoT's security measures and where it isn't working well enough right now [11].

Jin et al. (2018) Four months of data on traffic flow from a car detector on a city ring road were picked to be looked into further. A deeper learning network-based better SAE model is proposed to get the information from traffic flow data about features. A greedy layer-wise method is another way to train this model to guess. The predictions showed that this new model, which is more accurate, is the best way to guess how traffic will move [12].

Kaushik, Singh and Yadav (2018) Tensor Flow is the DL tool that was used to arrange the brain cells. A lot of high-level and mid-level APIs can be used to build the network. TensorFlow's Estimator API is used to build the neural network, and random data from the test sample is used to make the predictions. You can also use Adam optimizer to find the best loss function. This makes the model work about 98.6% to 99.8% of the time [13].

Huang, Chiang and Li (2017). Big data can be used to predict and understand mobile Internet traffic. This is the ground for smart management features as they move towards 5th generation (5G) cell phones. In it, forecasts are made about mobile traffic. Three of the most cutting-edge DL models being looked at at the moment are the RNN, the 3D CNN, and the CNN-RNN, which is a mix of the two. The tests show that CNN and RNN can separate geographical and time traffic factors. Beyond deep or non-DL methods, CNN-RNN is the most accurate model for all jobs, making estimates 70 to 80% of the time [14].

Table 1: Summary of literature review AI-Driven Cybersecurity for Industrial IoT Traffic Monitoring

Author	Dataset	Methods	Key Findings	Accuracy	Limitation/Gap
Dawoud, Shahristani and Raun (2019)	Not specified	Unsupervised DL (Autoencoder); Semi-supervised detection framework	Explores the potential of DL (esp. autoencoders) for anomaly detection in network traffic	Over 99%	No specific dataset; lacks real-time performance evaluation
Nagisetty and	UNSW-	MLP, CNN, DNN,	The proposed	Not specified;	Accuracy per model not

Gupta (2019)	NB15, NSL-KDD99	Autoencoder (via Keras DL Library)	framework effectively predicts malicious attacks across multiple DL models	metrics include accuracy, RMSE, and F1-score	individually detailed; lacks deployment details
Zolanvari, Teixeira and Jain (2018)	Not specified	Various ML algorithms for IIoT security	Highlights fundamental differences between IIoT and IT networks, requiring tailored ML approaches	High	Lacks concrete experimental results; more theoretical exploration
Jin et al. (2018)	City ringway traffic flow (4 months)	Improved Stacked Autoencoder (SAE); Greedy layer-wise training	The SAE model outperforms other methods in traffic flow prediction	High	Limited to urban traffic data; not network/cybersecurity-focused
Kaushik, Singh and Yadav (2018)	Random test dataset	TensorFlow-based Neural Network; Estimator API; Adam Optimizer	Neural network achieves high efficiency in prediction tasks	98.6% – 99.8%	No specific problem domain or dataset defined; limited to API-level experimentation
Huang, Chiang and Li (2017)	Big data on mobile internet traffic	RNN, 3D CNN, CNN-RNN	CNN-RNN outperforms others by capturing spatial (CNN) and temporal (RNN) features in traffic flow	70–80% forecasting accuracy	Lower accuracy compared to other DL methods; limited to traffic forecasting not security

3. Methodology

The proposed methodology utilizes the Bot-IoT dataset to enhance cybersecurity in Industrial IoT (IIoT) networks through AI-based traffic monitoring, with LSTM employed as the primary classification model. The process starts with preprocessing the data to make sure it is of good quality. This includes cleaning the data, dealing with timestamps, one-hot encoding, feature selection, and normalization to make the dataset best for training the model. There are two sets of data: training (80%) and testing (20%). The training set has been cleaned up first. An NB, an RF, and the suggested LSTM are some of the classification models that are used to find and sort different types of cyber risks. It is possible to rate how well a model works by its accuracy, precision, memory, and F1-score. Among these, the LSTM model achieves superior performance and is integrated into a real-time cybersecurity framework to enhance threat detection and monitoring capabilities in IIoT environments. The overall workflow of this methodology is illustrated in Figure 1.

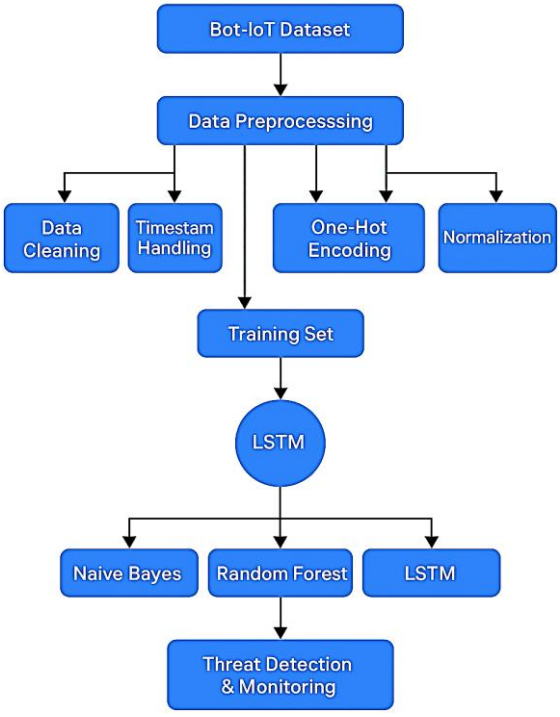

Fig 1: Methodology Flowchart of IoT Networks

3.1 Data Description

This approach was tested with the Bot-IoT dataset. Because this set of data includes both normal IoT network flow and different kinds of threats, you can see both. They picked this set because it shows a real IoT environment. Threats in this group include DoS and DDoS attacks, OS and service scans, keylogging, and data theft. At the network level, it has already been used to look for trends in the different kinds of data that devices send. If someone tries to break into the IoT Infrastructure, these trends can be used to find them.

3.2 Data Analysis and Visualization

The aim of the Bot-IoT collection is to make Industrial IoT (IIoT) networks safer. It has both normal network traffic and a wide range of cyber dangers, such as DDoS, ransomware, SQL injection, and port scanning. The dataset provides a balanced representation of various attack types, making it suitable for evaluating intrusion detection systems and improving cybersecurity in IIoT environments through AI-based traffic monitoring.

The overall steps of the Cybersecurity for IIoT network flowchart are provided below:

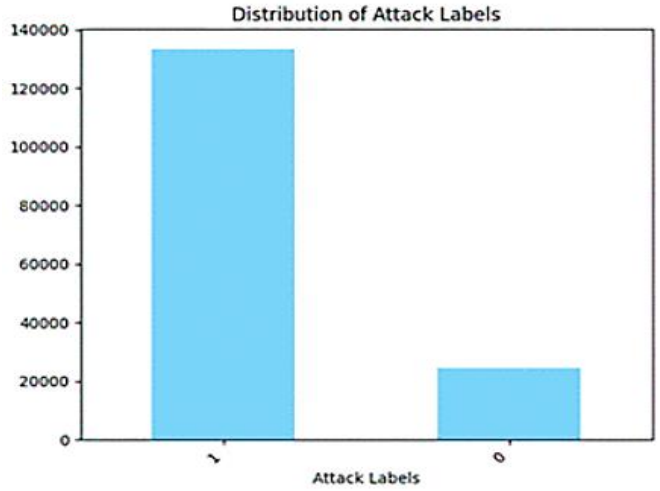

Fig 2: Bar graph of Bot-IoT dataset

The following Figure 2, Bar graph of Bot-Iot dataset the distribution of attack labels in a cybersecurity dataset, where '1' represents malicious traffic and '0' denotes benign traffic. There is an imbalance in the picture, with a much higher number of malicious samples. Such imbalance may affect ML models in which case, they might need resampling or class weight techniques to improve detection performance.

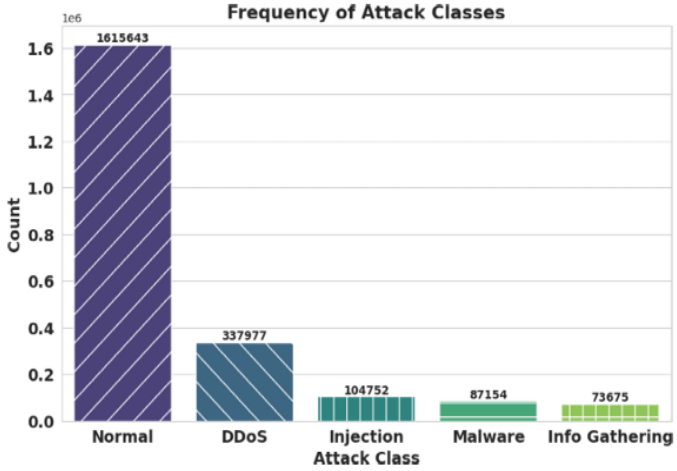

Fig 3: Frequency distribution of attack classes

Figure 3 displays the distribution of the Bot-IoT dataset's attack count. This shows that there is already a mismatch between the classes. Normal traffic makes up the majority of the dataset, with 1,615,643 instances, followed by DDoS attacks with 337,977. Injection attacks and malware come in at 104,752 and 87,154 instances, respectively, while information gathering attacks have the lowest count at 73,675. This imbalance shows how important it is to handle the data properly during model training, using methods like class weighting or data augmentation to make sure that the model learns from all attack categories in a fair and strong way.

3.3 Data Preprocessing

These steps are necessary to get the data ready for ML and DL algorithms and to make classification models work better. The balanced nature of the collection makes it hard to overfit. Using one-hot encoding, categorical class values, such as attack names, were turned into numerical representations. This made sure that they would work with ML models.

- Data cleaning: Remove records that are duplicates or don't belong You can deal with missing numbers by adding them in or taking them out. Eliminate any inconsistencies in data formatting
- Timestamp: The timestamp is changed into a datetime format. It also shows the Day, Hour, and Time that has passed since the mark.

3.4 One-Hot Encoding

One-hot encoding is a way to prepare data for ML algorithms by turning categorical factors into a number format that they can understand. It works by giving each unique category in a feature its own binary (0 or 1) column. This makes sure that classified data is stored in a way that is one-hot encoded to the Attack label.

3.5 Feature Selection

Feature selection methods try to improve classification performance by picking out only the most important features. This lets ML-based cybersecurity make better predictions, with each feature adding its own unique insights to improve accuracy. There are two main types of feature selection methods filter methods (like mutual information and association) and wrapping methods (like recursive feature removal).

3.6 Feature Normalization

Normalization is an important part of preprocessing data that can help classification systems work better and more accurately. A lot of different traffic values are in the data, and discretization by itself is not a good idea because the value range for discretization is wider than the normalization range. This process makes the data more accurate and consistent. A popular method is min-max normalization, which scales the numbers from 0 to Equation (1).

$$x_{normalized} = \frac{X - X_{\min}}{X_{max} - X_{min}} \quad (1)$$

X is the original feature, Xin is its lowest value, and Amax is its highest value.

3.7 Data Splitting

Data splitting is an important part of ML because it lets you test and train models by separating the information into separate sets. These researchers used 80% of the data to train their models and 20% to test them. The model could be tested and applied to many situations because of this.

3.8 Classification of LSTM Hybrid Model

LSTM cells are like the secret parts in recurrent networks because they have recurrent links. To what does x^t in the LSTM block refer? It's the message that was sent at time step t. At time step (t-1), ht-1 is the secret state, and ct-1 is the memory cell state. In this place, the block came from. The LSTM has gates for handling input, forgetting, and sending information. The following ways can help you find an LSTM's forget, input, output, and cell state gates. What data can and can't pass through the forget gate $f_i^{(t)}$ for cell i in time step t is controlled by the sigmoid activation function (π) in Equation (2). Here, b_i^f, Z_{ij}^f, and D_{ij}^f are the forget gates' deviation, input weight, and repeated weights, in that order. LSTM cell $n_i^{(t)}$ is shown in Equation (3), along with its current state. b, Z, and D are the deviation, input weight, and repeated weights that are coming into the cell, respectively. In Equation (4), it can see how to figure out the cell's input gate. This calculation is done in a way that is like how the forget gate calculation is done. In Equation (5), $s_i^{(t)}$ is the output gate and $h_i^{(t)}$ is the secret state. The solution for the output gate is shown in Equation (6) is defined as below:

$$f_i^{(t)} = \sigma\left(b_i^f + \sum jZ_{ij}^f x_j^{(t)} + \sum jD_{ij}^f h_j^{(t-1)} \right) \quad (2)$$

$$n_i^{(t)} = f_i^{(t)} n_i^{t-1} + P_i^{(t)} \sigma\left(b_i + \sum jZ_{ij}^f x_j^{(t)} + \sum jD_{ij}^f h_j^{(t-1)} \right) \quad (3)$$

$$p_i^{(t)} = \sigma\left(b_i^p + \sum jZ_{ij}^p x_j^{(t)} + \sum jD_{ij}^p h_j^{(t-1)} \right) \quad (4)$$

$$h_i^{(t)} = \tanh(n_i^{(t)}) s_i^{(t)} \quad (5)$$

$$s_i^{(t)} = \sigma\left(b_i^0 + \sum jZ_{ij}^0 x_j^{(t)} + \sum jD_{ij}^0 h_j^{(t-1)} \right) \quad (6)$$

b^0 stands for the deviation, Z^0 for the input weight, and D^0 for the repeated weights.

3.9 Performance Metrics

Several important metrics are used to rate how well their suggested model works for IIoT cybersecurity. They can learn something different about how useful the model is from each measure. It is necessary to comprehend the confusion matrix numbers in order to locate them. "True Positive," "True Negative," "False Positive," and "False Negative" are the four signs. The math for these steps looks like this:

- True positive (TP): the number of harmful codes that can be found.
- True negative (TN): the number of correctly known benign codes.
- False positive (FP): the number of times that harmless code was mistakenly labelled as malware.
- False negative (FN): the number of times that harmful code is mistakenly thought to be harmless code by a detector.

3.9.1 Accuracy

Divide the number of correctly predicted observations by the overall number of observations to get the accuracy [15]:

$$Accuracy = \frac{Number\ of\ currect\ prediction}{Total\ number\ of\ prediction} \quad (7)$$

Equation. (7) shows that the model can generally correctly classify both attack and normal activities as having a high level of accuracy.

3.9.2 Precision

A lot of precision is needed to cut down on erroneous results. The accuracy of your guesses can be judged by comparing the number of correctly predicted positive observations to the total number of expected positive observations:

$$Precision = \frac{True\ Positive}{True\ positive + False\ positive} \quad (8)$$

In cybersecurity, accuracy, as shown in Equation (8), is very important to make sure that real attacks are found and reported properly, lowering the chance of false alarms.

3.9.3 Recall

Makes sure the model can find all the real good cases:

$$Recall = \frac{True\ positive}{True\ positive + False\ negative} \quad (9)$$

Even though there is a chance of false positives, high recall, as shown in Equation (9), is necessary to find most strikes.

3.9.4 F1 Score

One way to find the F1-score is to take the harmonic mean of the Precision and Recall scores. It's a mix of Recall and Precision:

$$F1 - \text{score} = \frac{2 \times recall \times precision}{recall + precision} \quad (10)$$

There is a useful measure in Equation. (10 that includes both false positives and false negatives) That can be used when datasets aren't balanced.

3.9.5 ROC

The area under the ROC curve is found by comparing the True Positive Rate (Recall) to the False Positive Rate. This is known as AUC. In general, the AUC shows how well the model worked:

$$AU = \int_0^1 TPR(x)dx \quad (11)$$

As shown in Equation (11), a higher AUC means a better-performing model that can tell the difference between classes. To obtain an accurate assessment of how well a classification model works, these tests are very important. It's a way to figure out how well the plan works in general. What about precision, recall, and the F1-score? These tell you a lot about how well the model handles false positives and false negatives, as well as how well accuracy and memory work together. It depends on the problem being solved and the results that are wanted from the model review, which metrics to use.

4. Result Analysis and Discussion

Windows 7" was the operating system, "Jupiter Notebook" was the writing tool, and "Python" was the programming language. It had 16 GB of RAM and an "Intel Core" i3 CPU. A model named Bot-IoT dataset was tested for memory, accuracy, precision, and F1-score for hacks and IoT networks.

4.1 Experiment Results

At this point, they have the trial data with the model that was used for:

Table 2: LSTM model Performance on Bot-IoT dataset

Measure	LSTM
Accuracy	99.74
Precision	99.99
Recall	99.75
F1-score	99.87

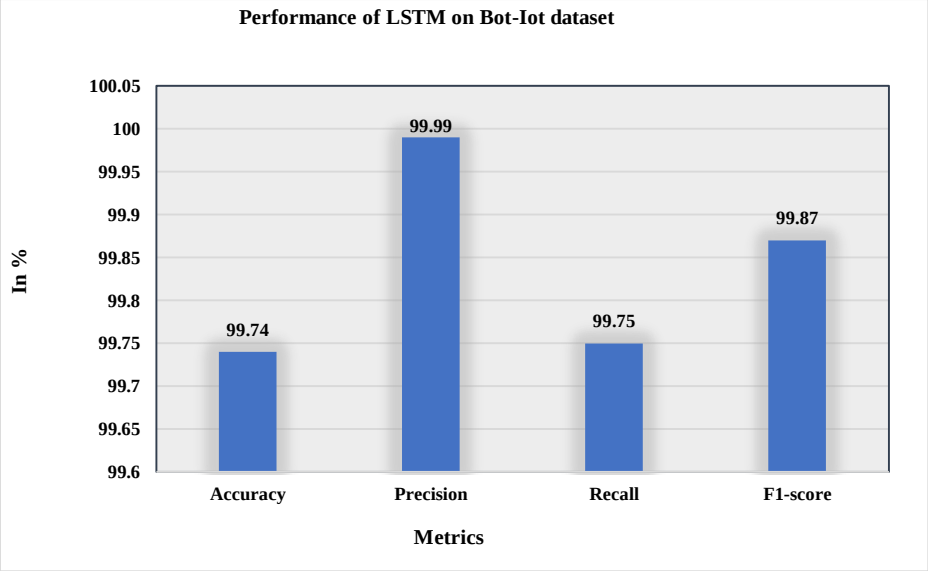

Fig 4: Bar Graphical representation of LSTM model

LSTM does well with some important rating features on the Bot-IoT dataset, as seen in Table II and Figure 4. A score of 99.87 on the F1 test means that the model was accurate 99.74 times, precise 99.99 times, remembered 99.75 times, and had a memory score of 99.75.With very few false positives, these results also showed that the LSTM model was good at finding bad things that were happening without giving up too much accuracy and recall. This high performance is shown in the bar graph of the Figure 4 which also demonstrates a strong and useful model for finding the intrusions in the IoT network settings.

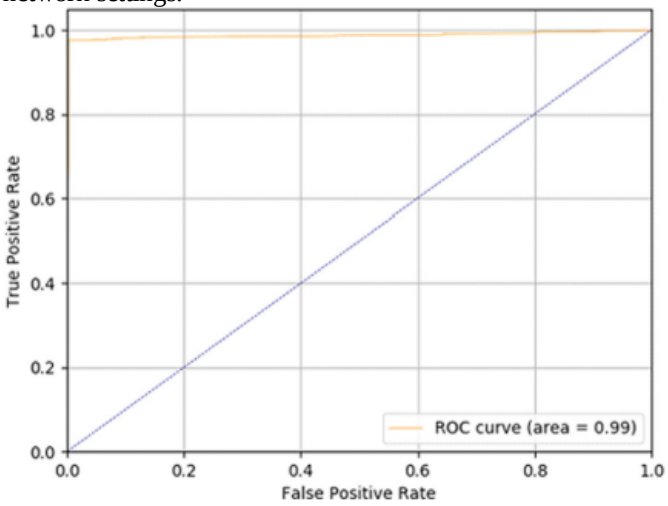

Fig 5: ROC curve of LSTM model

Figure 5 shows the ROC curve for the LSTM model that was tested on bot IoT. The plot of the curve closely follows the top left corner of the graph and greatly illustrates a near-perfect classification performance. Area Under Curve (AUC) of 0.99 is recorded that can distinguish malicious and benign traffic very well. There are a good number of both true positives and false positives in this model. It also means that there are fewer mistakes, which makes intrusion detection in IoT networks more reliable.

4.2 Comparative Analysis

This part compares IoT networks based on security using the Bot-IoT dataset. LSTM, RF, and NB models can be compared using performance measures like F1-score, memory, accuracy, and precision.

Table 3: Comparison between LSTM and Existing Model Performance

Measure	Accuracy
LSTM	99.74
Random forest[16]	92.67
NB[17]	79

A comparative analysis of the Accuracy of the LSTM Model versus Random Forest is shown in Table III. The LSTM Model versus NB is also presented. With a 99.74% success rate in the Bot-IoT dataset, the LSTM model does much better than the others, showing that it can accurately sort network traffic. However, NB model had a significantly lower accuracy of 79% whereas the Random Forest model with accuracy of 92.67%. In particular the LSTM model is proven to be successful and robust compared to the traditional approach in the problem of the IoT intrusion detection.

5. Conclusion and Future Work

ML methods can enhance China's Industrial IoT (IIoT) networks by making them safer and more reliable. ML helps Eliot systems put complex formulas to use, use data analysis, which enables them to find problems, avoid security risks, and run much smoother. With ML, IoT networks can properly search out and react to the possibility of hacking in real time. This paper was presented with a robust AI based framework to enhance the cybersecurity of Eliot network using DL techniques for traffic monitoring and classification with such framework. For preprocessing, analysis and classification of network traffic fed into the network traffic dataset, the proposed methodology has been implemented using Bot-IoT dataset. The Long Short-Term Memory (LSTM) model had the best accuracy score of 99.74%, the highest precision score of 99.99%, the highest memory score of 99.75%, and the highest F1 score of 99.87%. This result indicates that cyber threats can be well detected and classified using the model to a high extent compared to random forest and naïve bays models.

Future work based on hybrid DL models and ensemble models is an attempt to improve detection accuracy and reduction of false alarms. Additionally, to realize edge computing to its full potential for influencing real time for threat detection and classification, the latency should be kept at its minimum level. Investigating transfer learning and federated learning approaches can also provide opportunities for distributed, privacy-preserving, and adaptive cybersecurity solutions across heterogeneous Eliot infrastructures.

References

[1] H. Chen, M. Hu, H. Yan, and P. Yu, "Research on industrial internet of things security architecture and protection strategy," in *Proceedings - 2019 International Conference on Virtual Reality and Intelligent Systems, ICVRIS 2019*, 2019. doi: 10.1109/ICVRIS.2019.00095.

[2] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, "Rule Generation for Signature Based Detection Systems of Cyber Attacks in IoT Environments," *Bull. Networking, Comput. Syst. Softw.*, 2019.

[3] S. S. S. Neeli, "The Significance of NoSQL Databases: Strategic Business Approaches and Management Techniques," *J. Adv. Dev. Res.*, vol. 10, no. 1, p. 11, 2019.

[4] O. Ibitoye, O. Shafiq, and A. Matrawy, "Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks," in *Proceedings - IEEE Global Communications Conference, GLOBECOM*, 2019. doi: 10.1109/GLOBECOM38437.2019.9014337.

[5] A. Immadisetty, "Edge Analytics vs. Cloud Analytics: Tradeoffs in Real-Time Data Processing," *J. Recent Trends Comput. Sci. Eng.*, vol. 13, no. 1, pp. 42–52, 2016.

[6] S. S. S. Neeli, "Serverless Databases : A Cost-Effective and Scalable Solution," *IJIRMPS*, vol. 7, no. 6, 2019.

[7] M. Shafiq, X. Yu, A. K. Bashir, H. N. Chaudhry, and D. Wang, "A machine learning approach for feature selection traffic classification using security analysis," *J. Supercomput.*, 2018, doi: 10.1007/s11227-018-2263-3.

[8] J. Q. Gandhi Krishna, "Implementation Problems Facing Network Function Virtualization and Solutions," *IARIA*, pp. 70–76, 2018.

[9] A. Dawoud, S. Shahristani, and C. Raun, "Deep learning for network anomalies detection," in *Proceedings - International Conference on Machine Learning and Data Engineering, iCMLDE 2018*, 2019. doi: 10.1109/iCMLDE.2018.00035.

[10] A. Nagisetty and G. P. Gupta, "Framework for detection of malicious activities in IoT networks using keras deep learning library," in *Proceedings of the 3rd International Conference on Computing Methodologies and Communication, ICCMC 2019*, 2019. doi: 10.1109/ICCMC.2019.8819688.

[11] M. Zolanvari, M. A. Teixeira, and R. Jain, "Effect of imbalanced datasets on security of industrial IoT using machine learning," in *2018 IEEE International Conference on Intelligence and Security Informatics, ISI 2018*, 2018. doi: 10.1109/ISI.2018.8587389.

[12] Y. Jin, W. Xu, P. Wang, and J. Yan, "SAE Network: A Deep Learning Method for Traffic Flow Prediction," in *2018 International Conference on Information, Cybernetics, and Computational Social Systems, ICCSS 2018*, 2018. doi: 10.1109/ICCSS.2018.8572451.

[13] P. Kaushik, S. Singh, and P. Yadav, "Traffic Prediction in Telecom Systems Using Deep Learning," in *2018 7th International Conference on Reliability, Infocom Technologies and Optimization: Trends and Future Directions, ICRITO 2018*, 2018. doi: 10.1109/ICRITO.2018.8748386.

[14] C. W. Huang, C. T. Chiang, and Q. Li, "A study of deep learning networks on mobile traffic forecasting," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, 2017. doi:

10.1109/PIMRC.2017.8292737.

[15] O. Alkadi, N. Moustafa, B. Turnbull, and K. K. R. Choo, "Mixture localization-based outliers models for securing data migration in cloud centers," *IEEE Access*, 2019, doi: 10.1109/ACCESS.2019.2935142.

[16] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, "A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks," *Electron.*, 2019, doi: 10.3390/electronics8111210.

[17] J. Alsamiri and K. Alsubhi, "Internet of things cyber attacks detection using machine learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 12, pp. 627–634, 2019, doi: 10.14569/ijacsa.2019.0101280.

[18] Kalla, D., Kuraku, D. S., & Samaah, F. (2021). Enhancing cyber security by predicting malwares using supervised machine learning models. *International Journal of Computing and Artificial Intelligence*, *2*(2), 55-62.