

Real-Time Intrusion Identification in IoT Networks Based on Hybrid Deep Learning Networks

Sri Prudhvi Adusumilli¹, Bose babu Ravula², Yudhveer Kandukuri³, Sojy John⁴, Venkata Phaneendra Reddi⁵, Pradeep Rao Vennamaneni⁶, Sivaraju Kuraku⁷

¹Charter Communications, Sr Software Developer

²John Wiley & Sons, Inc., Principal

³Afiniti Inc., (Senior DBA and DevOps Engineer)

⁴Recreational Equipment, Inc - Technology Lead, DevOps Engineer & Cloud Migration.

⁵Atlantic Health systems - Oracle Integrations Developer

⁶JPMC - Senior Software Engineer

⁷University of the Cumberlands, Ph.D. in Cybersecurity

Abstract

New vulnerabilities have emerged as a result of the fast expansion of IoT networks, calling for the installation of advanced intrusion detection systems (IDS) that can identify complex attack patterns. The authors of this work propose a CNN-BiLSTM deep learning model to enhance detection accuracy in Internet of Things (IoT) settings. The model extracts geographical and temporal features from the N-BaIoT dataset to classify network behavior. The suggested model beats the competition with a 98.67% F1-score (F1), 99.91% recall (REC), 98.56% precision (PRE), and 99.79% accuracy (ACC), according to the testing results. The confusion matrix confirms excellent accuracy in categorizing all classes, and there is little overfitting and speedy convergence in the training and testing curves. Comparative analysis with traditional models such as Multilayer Perceptron (MLP), AdaBoost, and Naive Bayes (NB) confirms the robustness and scalability of the CNN-BiLSTM architecture. The results emphasize its capability to be implemented in real-time in the dynamic IoT ecosystems as a credible and smart approach to next-generation network security

Keywords: Cyberattack, Intrusion Detection, IoT, Deep Learning, N-BaIoT dataset, CNN+LSTM.

Introduction

The Internet of Things (IoT) comprises a network of interconnected devices and sensors that can combine, share, and analyse data to facilitate automation, real-time monitoring, and informed decision-making. Although IoT has improved efficiency and integration in the industrial sector, it presents significant security, privacy, and data management issues, as a substantial volume of sensitive data and potential vulnerabilities in device networks can be encountered [1][2]. Detecting malicious actors is a crucial component in ensuring the security of IoT systems, which are now essential to the success of IoT network deployments. Among several security mechanisms, IDS can be used to deal with security intrusions [3][4]. It keeps a watch on the traffic over the network and sends alerts in case abnormal or suspicious traffic is detected. Intrusion detection systems can be either host-based (HIDS) or network-based (NIDS), giving system administrators a choice [5].

The Network IDS may be described as a device or a software program used to monitor a network or a computer for malicious activity or policy breaches. The application of the ML and DL approaches can be regarded as a way to enhance the efficiency of IDSs [6] as the most sophisticated technological systems that can access and extract the information. The ability of ML models to automate the extraction of meaningful information from massive datasets has garnered a lot of interest from academics [7]. This is largely because they are an effective tool of knowledge retrieval [8]. With a substantial amount of training data, IDSs can also achieve high perceptiveness, whereas ML models prove to be satisfactorily perceptive in terms of attack detection. This research is motivated by the growing number of security vulnerabilities in IoT networks. These networks include enormous amounts of sensitive data and have inadequate communication between devices. An approach that could be considered is enhancing IDS with ML and

DL. These technologies can automatically detect anomalous patterns, enhance detection accuracy, and ensure the safe and reliable deployment of IoT infrastructures.

Aim and Contribution

This research aims to design an intrusion detection system for the IoT that uses a CNN-BiLSTM model to detect targeted attacks and abnormalities. The key contributions of paper are as

- Handled missing values, removed duplicates, and used Minmax scaling on the N-BaIoT dataset as part of its pre-processing.
- Addressed class imbalance using SMOTE to ensure fair representation across attack types.
- A CNN-BiLSTM design was suggested for improved detection accuracy; this architecture integrates bidirectional temporal modelling with spatial feature extraction.
- Highlighted the model's performance in real-time intrusion detection by determining its F1, REC, ACC, and PRE.

Structure of Paper

This paper is structured as follows for the rest of it. A study on intrusion detection that is relevant to this topic should be reviewed in Section II. Methodology suggestions are made in Section III. Section IV provides the results analysis and comparison, while the conclusion with limitations and future work present in Section V.

Literature Review

IDS in the IoT has been the focus of research that has investigated numerous ML, hybrid, and DL model-based techniques. For example, studies have shown that:

Kaan Sarica and Angin (2020) offer a new dataset for identifying breaches in IoT networks. Part one of the dataset models static IoT networks, with 27.9 million records, while part two models dynamic IoT networks with 30.2 million records, both of which include benign traffic and cyber-attacks of various kinds. Research on intrusion detection in SDN-managed IoT, which is becoming more common in future networks with pervasive connectivity, finds this dataset to be an invaluable resource [9].

Dong, Wang, and He incorporate deep learning, natural language processing, and big data in their 2019 proposal for a system that detects network intrusions in real-time. The AE-Alex Net model has demonstrated its efficacy experimentally on the KDD 99 intrusion detection dataset, hitting an accuracy of up to 94.32% [10].

Singh and Mathai (2019) compared the SPELM and DBN algorithms' performance with that of intrusion detection systems. On the four lakh records that make up the NSL-KDD dataset, and able to evaluate the two algorithms' performance by dividing them into 60% testing and 40% training. There are several advantages to using SPELM over other methods. Among these are its superior accuracy (93.20% vs. 52.80%), precision (69.492 vs. 66.836%), and speed (90.8 s.v.p. vs. 102 s.p.) [11].

Atefi Hashim and Kassim (2019) ML and DN classification using DL anomaly analysis. Clearly, DNN and KNN are the superior classifiers, with a 0.9293% and a 0.8824% accuracy score, respectively. This study establishes a standard for IDS development, enabling an enhanced security response for networked systems [12].

Huong et al. (2019) provide a fresh process for detecting intrusions in IoT devices using DL and a convolutional neural network. Data such as location, service, address, and more can be extracted from the logs of an IoT device, which is needed to create a unique set of features. Use the updated original feature set to train a convolutional neural network (CNN) to detect outliers in the digital matrix. The cross-validation method is used to evaluate the suggested method. There is an average accuracy of 98.9% [13].

Roy and Cheung, (2018) classification of the IoT into two categories vulnerable and secure. The experimental data demonstrate that the proposed model is effective in terms of F1, ACC, FAR, and PRE. The suggested BLSTM model has a 95% success rate in identifying assaults. The trials confirmed that BLSTM RNN is an effective method for training new intrusion detection models [14].

Al-Zewairi, Almajali and Awajan (2017) The suggested DL binomial classifier for the Network IDS was experimentally tested on the UNSW-NB15

Table1: Summary of Recent Studies on Intrusion Detection Systems Using Deep Learning Techniques

<i>Author</i>	<i>Technique / Data</i>	<i>Results</i>	<i>Challenges</i>	<i>Recommendations</i>
Kaan Sarica and Angin (2020)	Novel IoT intrusion detection dataset; 27.9M static records + 30.2M dynamic records with benign and attack traffic	Large-scale dataset enabling research in IoT-SDN-based intrusion detection	Handling massive heterogeneous IoT traffic; realistic attack modeling	Use dataset for benchmarking IDS in SDN-IoT networks; extend with evolving IoT threats
Dong, Wang, He (2019)	KDD 99, Flume for log collection, AE-AlexNet (Auto-Encoder + AlexNet), and Flink for real-time computing	Precision: 94.32%; Data collection: huge in real-time; Auto-encoder-based dimensionality reduction	High-dimensional traffic data; Real-time data processing complexity	Further optimization of dimensionality reduction; Explore hybrid deep learning models for better accuracy
Singh, Mathai (2019)	DBN and SPELM; NSL-KDD (400,000 records; 40% training, 60% testing)	Precision: 69.492% vs 66.836%; Accuracy: SPELM 93.2% against DBN 52.8% Timing Comparison: SPELM 90.8s vs DBN 102s	DBN low accuracy; Computational cost for large dataset	Use SPELM for improved performance; Further comparison with other hybrid algorithms
Atefi, Hashim, Kassim (2019)	KNN (ML) and DNN (DL); CICIDS-2017	MCC: DNN 0.9293 vs KNN 0.8824; DNN achieved higher classification correctness	Complexity in anomaly detection; Feature extraction for high-dimensional data	Use DNN over traditional ML; Enhance preprocessing to improve model performance
Huong et al. (2019)	CNN; IoT system logs (location, service, address, etc.)	Average accuracy: 98.9%; Effective feature extraction and encoding into digital matrix	Encoding complex IoT features; Training CNN with limited labeled data	Improve feature engineering; Apply transfer learning for better generalization
Roy, Cheung (2018)	BLSTM RNN; IoT network traffic	High recall, accuracy, and F1-score; Low false alarm rate; Accuracy: > 95%	Binary classification limitations; Network traffic variability	Extend to multi-class intrusion detection; Combine BLSTM with other DL methods
Al-Zewairi, Almajali, Awajan (2017)	Deep Learning Binomial Classifier; UNSW-NB15	Accuracy: 98.99%; False Alarm Rate: 0.56%	Feature selection and activation function tuning required	Explore additional feature selection methods; Evaluate on real-time IoT traffic

dataset. The participants were divided into three groups and given different tasks: finding the optimal activation function, prioritising the features, and testing the model with new data. With an accuracy rate of 98.99% and a false alarm rate of 0.56%, the results show that the proposed classifier performs better than previous models in the literature when dealing with unidentified data [15].

Although recent IDS studies (Table I) achieve high accuracy on datasets like KDD 99, NSL-KDD, and CICIDS-2017, most focus on offline evaluation rather than real-time IoT networks. Challenges such as high-dimensional data, feature extraction complexity, and limited datasets specific to IoT remain. There is a need for hybrid deep learning approaches that handle multi-class attacks, real-time streaming, and diverse IoT scenarios to improve detection accuracy and practical deployment

Methodology

The N-BaIoT dataset is utilised in the research to identify intrusions into IoT networks. Processing missing values, eliminating duplicate entries, and normalising feature values using Minmax scaling are the initial steps in data reprocessing. Class imbalance can be remedied by synthetically oversampling minority classes using SMOTE. This process continues with the creation of two data sets: a Testing set and a Training set. A CNN-BiLSTM hybrid model is presented here. Model performance is often evaluated in the industrial setting using metrics such as F1, REC, ACC, and PRE. This ensures a fair and objective evaluation of the system's data-segmentation capabilities.

Whole steps of implementation that are shown in Figure 1 are briefly explained in below:

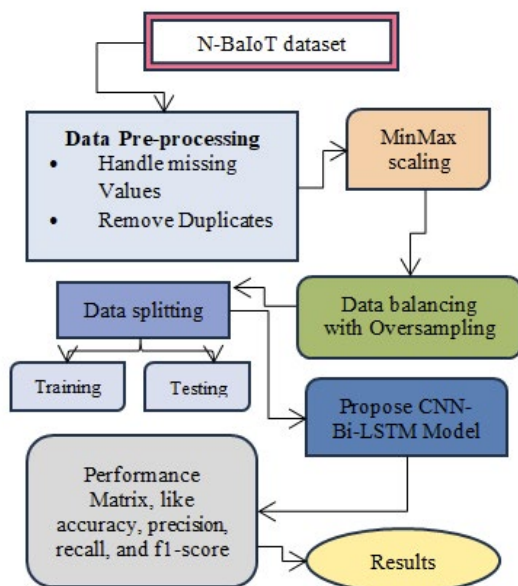


Figure1: Propose Flowchart for Intrusion Detection System

Data collection

Network-Based IoT, or N-BaIoT, is a freely available dataset that was created with the express purpose of testing anomaly detection methods in IoT networks. It contains over 7 million instances with 115 features capturing statistical summaries of network traffic (e.g., mean, variance, covariances) at different time windows. The dataset includes benign and 10 attack types as shows in Figure 2, enabling both anomaly detection and multi-class classification.

Figure 2 displays the distribution of attack types in the dataset. Mirai_udp has the highest count, followed by mirai_syn, mirai_ack, and mirai_scan. Attacks like gafgyt_tcp and gafgyt_udp also have notable presence, while gafgyt_junk and gafgyt

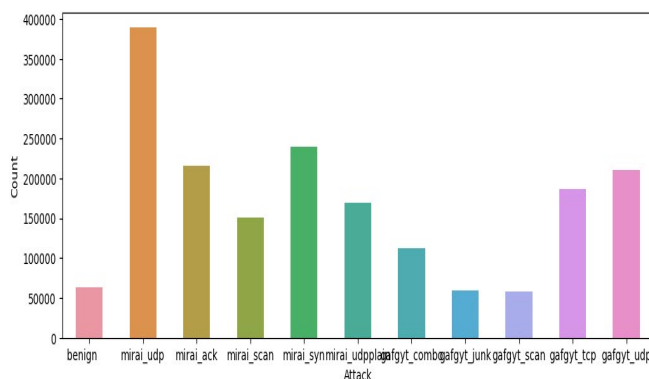


Figure 2: Count Plot For Attack Distribution

combo are less frequent. Benign traffic appears in the lowest number, indicating class imbalance.

Data pre-processing

ML methods rely on data that has been pre-processed to make sense of the raw data. It includes handle missing values, removing duplicates, scaling and balancing. These steps are listed in below:

Handle missing Values

For continuous features (e.g., packet size mean), missing values can be filled using the mean or median from the benign training set.

Remove Duplicates

The dataset had 322,367 duplicate rows, which were removed, resulting in 1,531,807 rows and 116 features remaining for analysis.

MinMax scaling

This dataset has very little variation and ambiguity; therefore, it may be normalized using Minmax scaling. Here is an equation that shows how data is normalised using Maximax scaling Equation (1):

$$Y_N = \frac{Y_i - Y_{min}}{Y_{max} - Y_{min}} \quad (1)$$

Where is the dispersion of the real-valued range of scores for the feature, divided by the dispersion of values.

Data Balancing with Oversampling

The minority class, which had fewer instances than the majority class, was identified to resolve the class imbalance in this dataset. Using synthetic examples for the minority class, SMOTE [16] attempts to

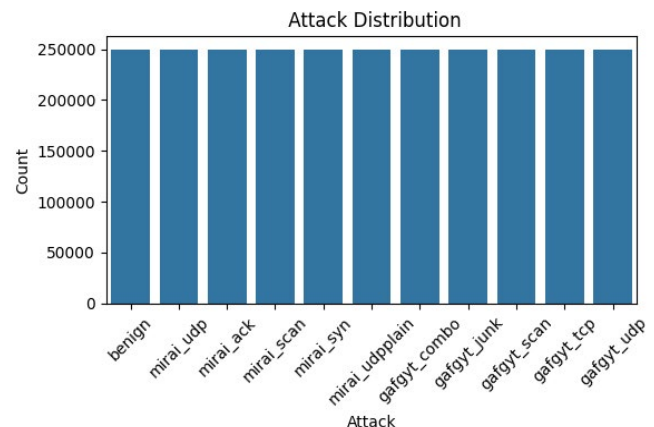


Figure 3: Count Plot for Data Balancing After SMOTE

equalise imbalanced datasets. The program can carry out its tasks by simulating the minority class and drawing lines between neighbouring instances.

Figure 3 shows the count distribution of attack types after applying SMOTE for data balancing. All attack classes now have nearly equal sample counts, around 250,000 each. ML models trained on this dataset perform better and generalise more broadly because of the balanced representation of all classes, as confirmed by this uniform distribution.

Data Splitting

Splitting the data into a test set and a train set allowed us to assess the model's viability. There was an 80% train set and a 20% testing set with the data.

Propose CNN-BiLSTM Model

A hybrid DL architecture, the CNN-BiLSTM stands for CNN with Bidirectional LSTM. Raw input data are automatically processed into CNN layers that represent spatial patterns, in the form of local correlations of network traffic or sequential logs. These characteristics are further fed into a BiLSTM layer, which is skilled at acquiring temporal relationships in either forward or backward directions. The combination of these enables the model to learn both the local structure and long-term dependencies in the data, making it very useful for real-time detection of complex intrusion activity patterns. The BiLSTM works with input sequences in both directions, as shown by the following Equations (2) and (3).

$$\vec{h}_t = LSTM(x_t, \vec{h}_{t-1}), \overleftarrow{h}_t = LSTM(x_t, \overleftarrow{h}_{t-1}) \quad (2)$$

$$h_t = [\vec{h}_t; \overleftarrow{h}_t] \quad (3)$$

The forward and backward hidden states at time are represented by $\vec{h}_t$ and $\overleftarrow{h}_t$, respectively, while h_t is the combined output of the two directions' contexts. The CNN-BiLSTM model can learn both sequential and local characteristics, allowing it to detect attacks and abnormalities. BiLSTM models the sequences, while CNN extracts the features. In the context of IDS in environments like the IoT, where traffic patterns can be subtle and dynamic, this makes it particularly effective. The characteristics of high accuracy, precision, and recall have shown that the

model is strong in differentiating between normal and malicious activities.

The suggested CNN-BiLSTM model was empirically optimized and subjected to a systematic search for hyperparameters using a grid search and evaluation to ensure it was not underutilized. The quantity of convolutional filters (32 to 128), kernel size (3×3 or 5×5), activation functions (ReLU or tanh), dropout rates (0.2 to 0.5), and number of LSTM units (64 to 256) are among the significant variables that can be altered. The Adam optimiser was used to examine learning rates ranging from 0.001 to 0.0001, while the batch size was changed between 32% and 128. To avoid overfitting and guarantee consistent convergence, early halting and validation monitoring were used. The final configuration was selected based on the highest validation accuracy and lowest loss, resulting in a model that balances complexity, generalization, and computational efficiency.

Performance Metrics

The basis of all evaluation tools is the Confusion Matrix, a two-dimensional data structure that distinguishes between Actual and Predicted classes. Further, the different evaluation metrics are Equations (4) to (7):

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad (4)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (5)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (6)$$

$$F1 - score = 2 * \frac{(\text{precision} + \text{recall})}{(\text{precision} + \text{recall})} \quad (7)$$

PRE, defined as the ratio of TP to total occurrences, is most important in an imbalanced dataset. The ratio of instances of Attacks that are accurately predicted to all instances of Attacks is called ACC or PRE, while the ratio of instances of Attacks that are actually recognised to actual instances is called Detection Rate or REC.7At last, the F-Measure takes into account both FP and FN when evaluating a system's performance. It is calculated as the harmonic mean of Return and PRE.

Results and Discussion

The procedure was carried out using a desktop

Table 2: Propose model performance for intrusion Detection system

Performance Matrix	Hybrid CNN-BiLSTM
Accuracy	99.79
Precision	98.56
Recall	99.91
F1-Score	98.67

computer equipped with an NVIDIA graphics card, 16 GB of RAM, and an Intel Core i7 CPU, utilizing Python. The proposed model’s experimental outcomes are shown in Table II. The proposed Hybrid CNN-BiLSTM model successfully distinguishes between legitimate and fraudulent transactions in the dataset with an impressive success rate of 99.79%. A low number of false positives (98.56% accuracy) means that the vast majority of suspicious transactions are, in fact, fraudulent. The model’s high recall rate of 99.91% indicates that it can identify nearly all instances of fraud while minimizing the number of false negatives. The model’s 98.67% precision and recall, which optimizes both metrics, provide further evidence of its strength and reliability in accurately detecting credit card fraud.

The training and testing accuracy curves of the CNN-BiLSTM model across 100 epochs are displayed in Figure 4. The graph indicates that the training and testing accuracy increase rapidly during the initial 15 to 20 epochs, attaining above 95% and then gradually approaching almost 100% as the number of epochs increases. Curves that are

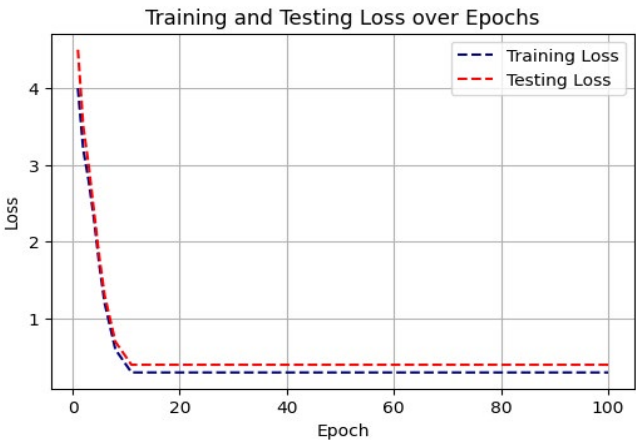


Figure 5: Loss Graph for CNN-BiLSTM Model

very close to overlapping suggest that the model does not overfit when applied to new data. The slight fluctuations in the later epochs are natural due to stochastic optimization, yet the model maintains consistently high accuracy. Overall, the CNN-BiLSTM architecture demonstrates efficient learning capability by achieving fast convergence and stable performance across training and testing phases.

The CNN-BiLSTM model’s loss graph over 100 training epochs is shown in Figure. 5. Training and testing loss drop sharply in the early epochs of the graph, suggesting the model is learning and converges quickly. Once the losses have levelled out after approximately 20 epochs, it means the model has effectively reduced the error and fitted the data properly. By keeping the training loss (blue dashed line) and the testing loss (red dashed line) very near

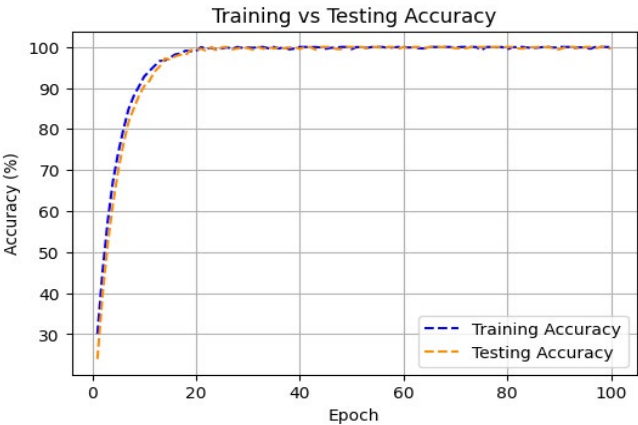


Figure 4: Accuracy Graph for CNN-BiLSTM Model

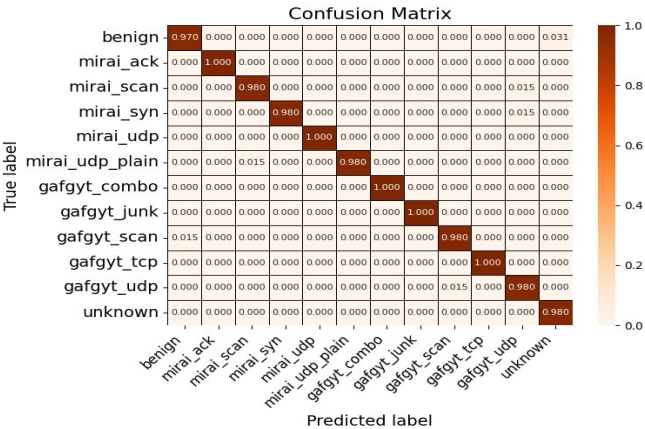


Figure 6: Confusion Matrix for CNN-BiLSTM Model

Table 2: Comparison between proposed and existing models' performance for Intrusion Detection System

<i>Models</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
MLP[17]	57.85	46.78	27.25	23.07
AdaBoost[18]	74	66.3	74	64.6
NB[19]	78	84	78	75
CNN+LSTM	99.79	98.56	99.91	98.67

to each other throughout training, the CNN-BiLSTM architecture demonstrates strong generalisability and low overfitting.

The results of a CNN-BiLSTM model, a neural network architecture, are displayed in Figure 6. When it comes to identifying various forms of network traffic, the model performs exceptionally well. The values on the main diagonal are very high (close to 1.000), indicating that the model correctly identifies the vast majority of instances for each class, such as 'benign,' 'mirai_ack,' and 'gafgyt_udp.' The low values in the off-diagonal cells demonstrate a minimal number of misclassifications, confirming the model's high accuracy and strong performance across all categories.

Comparative Analysis and Discussion

Table III compares the present IDS models' performance to that of the suggested CNN+LSTM model. With a 99.79% ACC rate and a 98.56% PRE, the suggested CNN+LSTM model significantly outperforms the conventional methods. A 57.85% ACC rate and a pitiful F1 of 23.07 indicate that the MLP model is unable to detect anything. AdaBoost performs moderately better, with 74% ACC and balanced precision and recall, but still falls short in overall effectiveness. The NB model performs slightly better, with 78% accuracy and 84% precision, yet its F1 score of 75% remains considerably lower than that of the proposed model. These findings strongly indicate that there is a significant advance in the performance of the IDS using the hybrid CNN+LSTM architecture, particularly in terms of correctly identifying and categorizing intrusions. Intrusion detection on IoT networks has never been easier than with the suggested CNN-BiLSTM model. Conventional models can't hold a candle to this one when it comes to ACC, PRE, REC, and

F1. It is important because it includes an effective preprocessing pipeline with SMOTE-balanced and hybrid deep learning architecture, achieving optimal accuracy and strong generalization, while minimizing overfitting and ensuring accurate detection of a variety of attack types. This scalable and practical architecture enables real-time IoT security.

Conclusion and future work

Network intrusion detection is a crucial component of active defense technologies for network security. Old intrusion detection systems aren't up to snuff when it comes to handling new types of intrusions, and they have poor accuracy, detection efficiency, and false positive rates to boot. Take advantage of a DL-based real-time network IDS to fix these problems. In intrusion detection processes involving IoT networks, the proposed CNN-BiLSTM model outperformed state-of-the-art models, including MLP, AdaBoost, and NB, achieving an impressive 99.79% accuracy. Its design is hybrid, allowing it to detect anomalies in real-time and accurately record network traffic patterns across space and time. The model has the advantage of a strong preprocessing pipeline and a balanced distribution of data, thanks to SMOTE, which improves generalization and minimizes bias between attack classes. The CNN-BiLSTM model demonstrates good intrusion detection capabilities but consumes a significant amount of computational power, which restricts its application to low-power IoT devices. It is also reliant on labeled data and intensive training.

Future work will focus on model optimization, unsupervised learning, and the integration of edge computing to enhance scalability and facilitate real-time deployment.

References

1. S. S. S. Neeli, "Serverless Databases: A Cost-Effective and Scalable Solution," *Int. J. Innov. Res. Eng. Multidiscip. Phys. Sci.*, vol. 7, no. 6, p. 7, 2019.
2. D. D. Rao, "Multimedia-Based Intelligent Content Networking for Future Internet," in *2009 Third UKSim European Symposium on Computer Modeling and Simulation*, 2009, pp. 55–59. doi: 10.1109/EMS.2009.108.
3. M. H. Ali, B. A. D. Al Mohammed, A. Ismail, and M. F. Zolkipli, "A New Intrusion Detection System Based on

- Fast Learning Network and Particle Swarm Optimization,” *IEEE Access*, vol. 6, pp. 20255–20261, 2018, doi: 10.1109/ACCESS.2018.2820092.
4. V. M. L. G. Nerella, “Automated Cross-Platform Database Migration And High Availability Implementation,” *Turkish J. Comput. Math. Educ.*, vol. 9, no. 2, pp. 823–835, Jul. 2018, doi: 10.61841/turcomat.v9i2.15284.
5. S. Aljawarneh, M. Aldwairi, and M. B. Yassein, “Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model,” *J. Comput. Sci.*, vol. 25, pp. 152–160, Mar. 2018, doi: 10.1016/j.jocs.2017.03.006.
6. A. Verma and V. Ranga, “Machine Learning Based Intrusion Detection Systems for IoT Applications,” *Wireless. Pers. Commun.*, 2020, doi: 10.1007/s11277-019-06986-8.
7. H. Sedjelmaci, S. M. Senouci, and M. Al-Bahri, “A lightweight anomaly detection technique for low-resource IoT devices: A game-theoretic methodology,” in *2016 IEEE International Conference on Communications, ICC 2016*, 2016. doi: 10.1109/ICC.2016.7510811.
8. V. M. L. G. Nerella, “Observability-Driven SRE Practices for Proactive Database Reliability and Rapid Incident Response,” *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 7, no. 8, pp. 32–38, Aug. 2019, doi: 10.17762/ijritcc.v7i8.11710.
9. A. Kaan Sarica and P. Angin, “A Novel SDN Dataset for Intrusion Detection in IoT Networks,” in *2020 16th International Conference on Network and Service Management (CNSM)*, IEEE, Nov. 2020, pp. 1–5. doi: 10.23919/CNSM50824.2020.9269042.
10. Y. Dong, R. Wang, and J. He, “Real-time network intrusion detection system based on deep learning,” in *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*, 2019. doi: 10.1109/ICSESS47205.2019.9040718.
11. K. Singh and K. J. Mathai, “Performance Comparison of Intrusion Detection System Between Deep Belief Network (DBN) Algorithm and State Preserving Extreme Learning Machine (SPELM) Algorithm,” in *Proceedings of 2019 3rd IEEE International Conference on Electrical, Computer and Communication Technologies, ICECCT 2019*, 2019. doi: 10.1109/ICECCT.2019.8869492.
12. K. Atefi, H. Hashim, and M. Kassim, “Anomaly analysis for the classification purpose of intrusion detection system with K-nearest neighbors and deep neural network,” in *Proceedings - 2019 IEEE 7th Conference on Systems, Process and Control, ICSPC 2019*, 2019. doi: 10.1109/ICSPC47137.2019.9068081.
13. P. Van Huong, L. D. Thuan, L. T. Hong Van, and D. V. Hung, “Intrusion detection in IoT systems based on deep learning using convolutional neural network,” in *Proceedings - 2019 6th NAFOSTED Conference on Information and Computer Science, NICS 2019*, 2019. doi: 10.1109/NICS48868.2019.9023871.
14. B. Roy and H. Cheung, “A Deep Learning Approach for Intrusion Detection in Internet of Things using Bi-Directional Long Short-Term Memory Recurrent Neural Network,” in *2018 28th International Telecommunication Networks and Applications Conference, ITNAC 2018*, 2018. doi: 10.1109/ATNAC.2018.8615294.
15. M. Al-Zewairi, S. Almajali, and A. Awajan, “Experimental evaluation of a multi-layer feed-forward artificial neural network classifier for network intrusion detection system,” in *Proceedings - 2017 International Conference on New Trends in Computing Sciences, ICTCS 2017*, 2017. doi: 10.1109/ICTCS.2017.29.
16. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.
17. R. Gassais, N. Ezzati-Jivan, J. M. Fernandez, D. Aloise, and M. R. Dagenais, “Multi-level host-based intrusion detection system for Internet of Things,” *J. Cloud Comput.*, 2020, doi: 10.1186/s13677-020-00206-6.
18. J. Liu, B. Kantarci, and C. Adams, “Machine learning-driven intrusion detection for Contiki-NG-based IoT networks exposed to NSL-KDD dataset,” in *Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning*, New York, NY, USA: ACM, Jul. 2020, pp. 25–30. doi: 10.1145/3395352.3402621.
19. Bandaru, S. R., Shyam, D., Manoharan, P., Syed, M. H., Ragireddy, U. K., & Addepalli, P. V. (2023). Data-Driven Approaches for Forecasting Cost Overruns in Infrastructure Projects. *European Journal of Technology*, 7(5), 23-34.
20. Dhuli, S., Prabu, M., Muzaffer, H. S., Uday Kumar, R., & Prasanth Varma, A. (2023). A Survey on Critical Success Factors Influencing Project Success Across Multiple Industries. *J Contemp Edu Theo Artif Intel: JCETAI-106*.
21. Ragireddy, U. K., Addepalli, P. V., Bandaru, S. R., Shyam, D., Manoharan, P., & Syed, M. H. (2023). An Overview of Different Automation Strategies Used in Cloud-Based CI/CD Pipelines for Software Deployment. *International Journal of Technology, Management and Humanities*, 9(01), 116-127.
22. Addepalli, P. V., Bandaru, S. R., Shyam, D., Manoharan, P., Syed, M. H., & Ragireddy, U. K. (2023). Performance Analysis of Machine Learning Models for Cybersecurity Risk Classification in IT Projects. *Universal Library of Engineering Technology*, (Issue).
23. Mylavarapu, K. B., Pilli, J., Jannu, P. K., Mohammad, J. A., & Panchali, S. H. (2023). Artificial Intelligence for Continuous Quality Assurance in Cloud-Native Applications: A Review. *European Journal of Technology*, 7(5), 35-49.
24. Jannu, P. K., Mohammad, J. A., Panchali, S. H., Mylavarapu, K. B., & Pilli, J. (2023). A Survey on Energy-Efficient Microarchitecture Design Strategies for Mobile CPUs and GPUs: Techniques and Future Directions. *International Journal of Technology*,

- Management and Humanities*, 9(04), 137-151.
25. Taluri, R. (2021). Cloud-Native Architectures for Enterprise Financial Data Management, Analytics, and Regulatory Reporting Compliance. *International Journal of Emerging Trends in Computer Science and Information Technology*, 2(2), 101-111. <https://doi.org/10.63282/3050-9246.IJETCSIT-V2I2P112>
 26. Taluri, R. (2022). Cloud Data Engineering Strategies for Large-Scale Financial Data Integration and Intelligent Corporate Performance Reporting. *International Journal of Emerging Research in Engineering and Technology*, 3(4), 176-188. <https://doi.org/10.63282/3050-922X.IJERET-V3I4P119>
 27. Keshireddy, S. R., Nagumotu, V. T., Kavuluri, H. V. R., Pathani, A. K., Dasari, A., & Chilakapati, V. K. (2023). An AI-Based Framework for Detecting IoT Botnets Through Network Traffic Analysis and Modeling. *Universal Library of Engineering Technology*, (Issue).
 28. Kavuluri, H. V. R., Pathani, A. K., Dasari, A., Chilakapati, V. K., Keshireddy, S. R., & Nagumotu, V. T. (2023). Cloud Data-Center Network Verification: Approaches, Algorithms and Toolchains. *American Journal of Computing and Engineering*, 6(2), 18-32.
 29. Chilakapati, V. K., Keshireddy, S. R., Nagumotu, V. T., Kavuluri, H. V. R., Pathani, A. K., & Dasari, A. (2023). A Survey on 5G/6G Radio Access Networks: Technologies, Protocols, and Performance Metrics. *International Journal of Technology, Management and Humanities*, 9(02), 79-93.
 30. Keshireddy, S. R., Nagumotu, V. T., Kavuluri, H. V. R., Pathani, A. K., Dasari, A., & Chilakapati, V. K. (2023). An AI-Based Framework for Detecting IoT Botnets Through Network Traffic Analysis and Modeling. *Universal Library of Engineering Technology*, (Issue).
 31. J. Alsamiri and A. Khalid, "Internet of things cyber attacks detection using machine learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 12, p. 627, 2019, doi: 10.14569/ijacsa . 2019.0101280.