Development And Testing of a Vulnerability Identification Model That Uses Deep Learning Model in A Healthcare IoT Architecture

CHIOMA JULIET[1], OZIOKO EKENE FRANK[2], EZUGWU LILIAN MARTINA[3]
[1]*Enugu State Polytechnic Iwollo*
[2]*Enugu State University Science and Technology*
[3]*Enugu State College of Education (Technical), Enugu*

Abstract- ***As a result of the speedy incorporation of Internet of Things (IoT) technologies in healthcare systems, it has managed to enhance service delivery, but also accelerated the appearance of potential cyber threats that jeopardize the safety of the patients, their data confidentiality, as well as the working dependability of the systems. This paper proposes the development and testing of a vulnerability identification model that uses deep learning model in a healthcare IoT setting. Three methods of deep neural networks (DNN) models, which comprise Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and a combination of CNN+LSTM models, were trained and evaluated on real-life vulnerability data acquired at the University of Nigeria Teaching Hospital. Important performance vector was evaluated to test the models: loss, accuracy, precision, recall, and F1-score. The performance of CNN+LSTM hybrid model as evaluated revealed that it yielded the best result in all the metrics with an accuracy of 97 percent in training, 93 in validation, 94 percent precision, 95 percent recall and a f1-score of 94 percent. This would mean that there is better ability to identify and categorize vulnerabilities together with reduced false positives and anomalies. The conclusion points to the promise of hybrid deep learning methods in pursuing improved security in healthcare IoT systems in terms of real-time and reliable detection and response capabilities of threats.***

Indexed Terms- ***Healthcare; Internet of Things; Vulnerability; Deep Learning; CNN; LSTM***

I. INTRODUCTION

Cyber-physical Systems (CPS) are the integration of cyber elements, networking, and physical systems to facilitate the exchange of information in real time (Parades et al., 2024; Yuan et al., 2024). CPS emerges from the integration of embedded computer and communication technologies into physical systems to automatically monitor and control processes effectively (Abdelrahman et al., 2024). Also, CPS are vital infrastructures such as the Internet of Things, industrial automation, healthcare, and smart cities. It blends physical components such as actuators, sensors, and machines with digital components such as control and monitoring systems to enhance optimal performance, thus making them the backbone of modern smart infrastructure (Segovia-Ferreiraet al., 2023). However, it has been reported severally that CPS are prone to various vulnerability issues, both in the physical layers and digital layers (Umer et al., 2022; Parades et al., 2024).

Umer et al. (2022) grouped the components of CPS into three, namely the physical layer, application layer, and network layer. According to Khan et al. (2022) and Umer et al. (2022), CPS during the design stage does not integrate security measures like other networking systems due to its heterogeneous nature, different operating protocols, software and hardware it uses during communication, thus making it vulnerable to cyber threats. CPS are even more vulnerable due to the connectivity with massive components and are prone to attack on each layer (Markakis et al., 2019). In the network layer, denial of service, flood attacks, signal jamming, and Sybil

attacks are among the major threat issues reported due to vulnerabilities in the layers(Khan et al. 2022).
One of the main applications of CPS is in healthcare, where it is used to integrate digital systems with physical healthcare devices to improve real-time data processing, control, and monitoring (Vyas and Bhargava, 2021). These systems include medical equipment that gathers and sends patient data to healthcare service providers through robotic surgical systems, wearable health monitors, and telemedicine platforms (Iqbal et al., 2020). These systems' connectivity allows for remote consultations, rapid interventions, and continuous vital sign monitoring, all of which contribute to optimal patient care. However, the reliance of the healthcare CPS (H-CPS) on different technologies to operate also presents the risk of vulnerabilities, such as those related to data breaches, illegal access, and system failures, all of which can have detrimental effects on patient privacy and safety (Xu et al., 2019).Vulnerability management refers to the process of identifying, classifying, and reporting this security vulnerability in the systems (Northern et al., 2021). In the H-CPS context, vulnerability detection and control aims to identify flaws in the different service layers and report to help mitigate cyber-attacks. Currently, to successfully manage vulnerability in H-CPS is very difficult due to the complex nature of its architecture(Knowles et al., 2015; Bernieri et al., 2018).

In the past, basic security frameworks like intrusion detection systems, firewalls, and patch management systems were some of the popular approaches for vulnerability management, but despite their success, they were often reactive and unable to provide real-time security assurance for general CPSs. Recently, Deep Neural Networks (DNN) have resonated as a powerful tool for real-time data analysis of complex patterns, making them suitable for pro-active vulnerability detection and control in CPS (Khazraei et al., 2022; Ashraf et al., 2022); However, there is limited work on the application of Deep Leaning (DL) for vulnerability management in CPS, thus necessitating the need for this work. In addition, it is rare; a work that considered vulnerability in the three layers of CPS, and finally zero-day vulnerability, has not been addressed in the CPS system.

This study designs a real-time vulnerability management model for electronic healthcare cyber-physical system using deep neural network.This was achieved through careful consideration of health care CPS as the case study and developing a model that ensures that information in patient health records is secured at all times.

II. METHODOLOGY

The spiral model will be employed for this work, offering an iterative and risk-driven approach to project management. It combines elements of both incremental development and systematic risk management, making it suitable for complex, evolving systems. Each phase, or loop, involves four key steps: planning and requirement gathering, risk analysis, development and testing, and evaluation. The project begins with basic requirements, and as the loops progress, the system is gradually refined and expanded. At each loop, risks are assessed and mitigated before moving to the next phase, ensuring that potential issues are addressed early. This iterative model allows for continuous refinement based on feedback, adaptability to changes, and a focus on minimizing project risks, ensuring robust and secure system development.

2.1 Data Collection

The data used for this work was collected from the University of Nigeria Teaching hospital, Enugu state Nigeria as the primary data source. The data include common vulnerabilities in the Intensive Care Unit (ICU) of the hospital, capturing key security attributes across the transport, network, and application layers of connected medical devices from 2019 to 2022. It includes structured records of known vulnerabilities, identified by `CVE ID` and categorized under `CWE ID` to specify the weakness type. Each entry provides details such as ` of Exploits` to indicate exploitation likelihood, `Vulnerability Type(s)`, and `Publish Date` with `Update Date` for tracking disclosure timelines. The `Score` (CVSS) quantifies severity, while `Gained Access Level`, `Access Complexity`, and `Authentication` highlight exploit difficulty. The dataset further assesses security impact through `Confidentiality Impact`, `Integrity Impact`, and

`Availability Impact`, mapping risks to patient data privacy, device reliability, and system uptime. This comprehensive structure enables predictive modelling for risk assessment and proactive mitigation of cybersecurity threats in critical healthcare infrastructures. The sample size of the data collected is 107606 features of vulnerability. The Table 1 presents the data description.

Table 1: Data description of Healthcare IoT vulnerability

Attribute	Data Type	Description
CVE ID	String	Unique identifier for the vulnerability (e.g., CVE-2023-XXXX).
CWE ID	String	Common Weakness Enumeration (CWE) identifier for the vulnerability type.
of Exploits	Integer	Number of known exploits available for the vulnerability.
Vulnerability Type(s)	String	Type of vulnerability (e.g., SQL Injection, Buffer Overflow).
Publish Date	Date (YYYY-MM-DD)	The date when the vulnerability was publicly disclosed.
Update Date	Date (YYYY-MM-DD)	The most recent update date of the vulnerability record.
Score	Float (0.0 - 10.0)	CVSS (Common Vulnerability Scoring System) score indicating severity.
Gained Access Level	String	The level of access gained if exploited (e.g., Admin, User).
Access Complexity	String (Low/Med/High)	Difficulty of exploiting the vulnerability.
Authentication	String (Required/Not Required)	Whether authentication is needed for exploitation.
Confidentiality Impact	String (Low/Med/High)	Impact on data confidentiality if exploited.
Integrity Impact	String (Low/Med/High)	Impact on data integrity if exploited.
Availability Impact	String (Low/Med/High)	Impact on system availability if exploited.
Description	String	A brief summary of the vulnerability and its impact.

2.2 Data Preparation

The collected data were processed using visualization, normalization and balancing approach. First the data structure was visualized in excel form to check for missing and duplicate values. This was done carefully using manual physical inspection by the researcher. The outcome showed that the data has no duplicate and missing values, and in addition all the features were observed to be numeric apart from the unique identifier. Data normalization was applied for dimensionality reduction using resampling approach based on Min-Maxscaler technique as shown in Equation 1 (Khalid et al., 2024).

$$X_{i,j} = (X_{i,j} - Min(X_j))/(Max(X_j) - Min(X_j)) \quad (1)$$

Where Xj represents the features of the j-th credit card transactions, X_(i,j) is the features X_j of sample i. for the data distribution of target variables. The imbalance data structure prompt the need for class balancing. This was address by adopting the random under sampling approach in Khalid et al. (2024). The algorithm was presented using the relationship between fraud transaction subset defined as D^1, legitimate subset defined as D^0 and D^' which represents under sample dataset.

Algorithm 1: Random class under sampling algorithm
$Input: D = DD^o \; U \; D^1$
$D' = D^1$
$For \; j = 1,2, \ldots \ldots \ldots card \; D^1 \; do$
$Select \; random \; sample \; x \in D^0$
$D' = D'U\{x\}$
$Delete \; x \; from \; D^0$
$End \; for$
$Output \; D'$

2.2 The Convolutional neural network (Benchmark CNN)

A CNN is a type of feedforward neural network made of four major blocks which are the input layer, convolutional layer, fully connected layer and output layer. The input layer is responsible for dimensioning input transaction data based on the scaled output from Equation 1, the convolutional layer takes care of data extraction process using filters and pooling functions, while the fully connected layer is where the training process takes place and finally the output layer which produces the final results. The Figure 1 presents the architecture of the CNN;

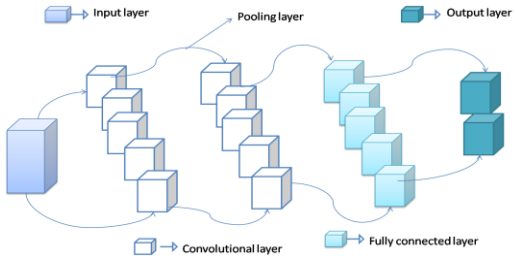

Figure 1: Architecture of CNN (Wang et al., 2024)

Figure 1 showed the architecture of the benchmark CNN which was adopted for this study. This convolutional layer used convolutional filters to extract local features, and with the help of nonlinear activation function produced similar output which forms input to the next convolutional layer. The mathematical model of CNN is defined by Wang et al. (2024) as Equation 2;

$$X_j^l = f\left(\sum_{i=1}^{M} x_j^{(l-1)} * k_{jk}^l + b_j^i\right) \quad (2)$$

Where X_j^lrepresents jth features of thel-$h$layer, f defines the activation function; M number of feature maps,i represent it$h$ feature maps in the l-k; while $k_{jk}^l$is the convolutional kernel and b_j^i is the offset. The convolutional layer here is a 1D convolution which has filter, kernel and rectified linear unit activation function defined as Equation 3. The filters are kernel are used for the convolutional scan, ensure that the matrix of the feature vectors are identified during the convolutional process.

$$f(x)=\max[0, (0, x)] \quad (3)$$

The function of the Equation 3 is to help address vanishing problem during the feature extraction process, introduction of nonlinearity to facilitate learning of more complex transaction patterns and also speed up the convergence of neurons (Zhou et al., 2017). During this convolutional process to map out feature vectors of the input transactional data, the pooling layers are applied to extract it. Several pooling techniques existing for this process such as the mean, maximum pooling technique, however Wang et al. (2024) revealed that the mean pooling process suffer limitation of spatial information loss, which can affect the quality of extracted features.

Based on this insight this work applied the maximum pooling techniques for the extraction process based on downsampling approach defined in the Equation 4;

$$X_j^i = f\ (down\ \left(x_j^{(l-1)}\right) + b_j^i \qquad (4)$$

Where f is the pooling function, down (.) mean the downsampling function while X_j^i is set to 0. The Table2 presents the architecture of the benchmark CNN, showing the number of layers, their functions and hyper-parameter sizes at each section.

Table 2: Architecture of the Benchmark CNN with output sizes

Layers	Output size	Hyper-parameters
1D convolution	[None; 29, 64]	256
Filter size	[32, 128]	32
Kernel	1	
Batch normalization	[None; 29, 64]	256
1D convolution (Layer 1)	[None; 29, 128]	24704
Filter size	[32, 128] kernel =1	
Activation function	ReLU	
Batch normalization (Layer 1)	[None; 29, 128]	512
Maximum pooling	[None; 14, 128]	0
1D convolution (layer 2)	[None; 14, 128]	0
Filter size	[32, 128] kernel =1	
Activation function	ReLU	
Batch normalization (Layer 2)	[None; 14, 256]	98560
Flatten	[None; 14, 256]	1024
Dense	[None; 3584]	1835520
Dropout	[None; 512]	0
Dropout factor	0.3; 0.8	
Dense	[None; 512]	513

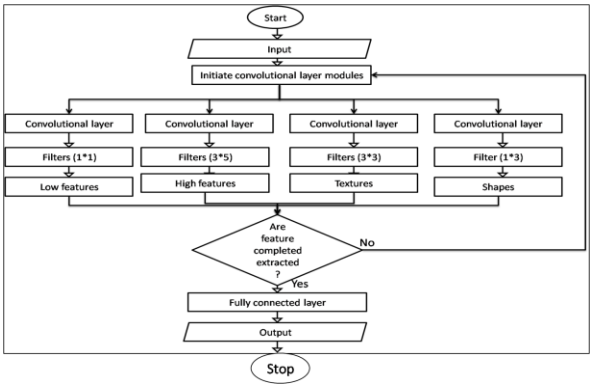

Figure 2: Flowchart of the CNN Model

2.3 Long Short-Term Memory (LSTM)

LSTM is a type of Recurrent Neural Network (RNN) designed to handle sequential data while overcoming the vanishing gradient problem. Unlike standard RNNs, LSTMs use memory cells with gates (input, forget, and output gates) to selectively store, update, and retrieve information over long time steps. This allows LSTMs to effectively learn dependencies in time-series data, such as detecting patterns in cybersecurity vulnerabilities, predicting network intrusions, or analysing ICU device logs for anomalies. The forget gate decides what past information to discard, the input gate determines which new information to store, and the output gate controls what part of the memory is passed to the next step. LSTMs are widely used in predictive modelling, anomaly detection, and cybersecurity threat analysis, making them ideal for analysing sequential attack patterns in healthcare networks. Figure 3 presents the flowchart of the LSTM.

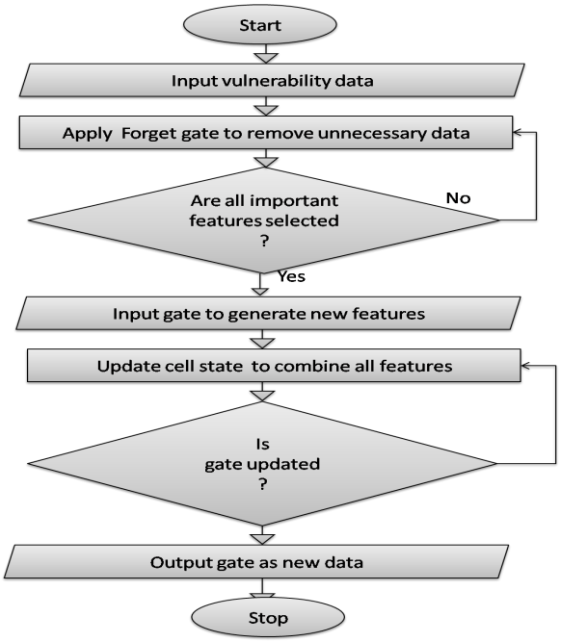

Figure 3: Flow chart of the LSTM

2.3 CNN-LSTM MODEL

A CNN-LSTM model is a hybrid deep learning architecture that combines CNN for spatial feature extraction and LSTM networks for capturing temporal dependencies in sequential data. CNN layers first process raw data, identifying key spatial patterns, while LSTM layers analyze the extracted features over time to detect trends, anomalies, or cyber threats. This model is highly effective for intrusion detection, vulnerability assessment, and anomaly detection in healthcare systems, enabling real-time security monitoring by identifying sequential attack patterns and abnormal behaviours in connected medical devices. Figure4 presents the flow chart of the CNN+LSTM.

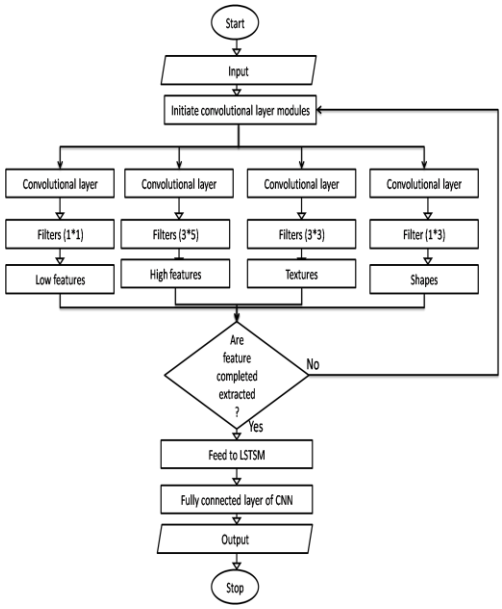

Figure 4: Flowchart of the CNN+LSTM

2.4 Training of the Deep Neural Network Models

The training process of the DNN models involved several key steps to ensure optimal performance in detecting vulnerabilities. First, the dataset containing common vulnerability data, was pre-processed by handling missing values, normalizing numerical features, and encoding categorical variables where necessary. Next, the dataset was split into training and testing subsets to evaluate the model's generalization capability. Data augmentation techniques were applied where necessary to balance the dataset and improve robustness. Feature extraction was performed using the CNN component, which captured spatial patterns, while the LSTM component processed sequential dependencies, allowing the model to retain temporal correlations.

Once the data preparation was complete, the model was trained using an optimized configuration of hyper parameters, including learning rate, batch size, and the number of layers. The Adam optimizer was selected for efficient gradient updates, and the categorical cross-entropy loss function was used to measure performance. The training process involved multiple epochs, with validation at each step to monitor overfitting. Dropout and batch normalization techniques were applied to improve generalization. Finally, model performance was evaluated using key metrics such as accuracy, precision, recall, and F1-

score to assess its effectiveness in identifying vulnerabilities within the dataset.

III. SYSTEM IMPLEMENTATION

The system implementation was carried out using Python, leveraging its extensive libraries and frameworks for deep learning and data processing. The implementation began with data preprocessing using NumPy and Pandas to clean and structure the dataset. Scikit-learn was used for feature scaling and dataset splitting into training and testing sets. The TensorFlow and Keras libraries were employed to design and train the hybrid deep neural network model, integrating CNN for feature extraction and LSTM for sequential pattern recognition. The training phase involved defining the network architecture, setting hyperparameters, and using the Adam optimizer for efficient weight updates. The model was trained in multiple epochs with validation at each stage to monitor performance and prevent overfitting using dropout and batch normalization techniques. Finally, the trained model was evaluated on test data using metrics such as accuracy, precision, recall, and F1-score. The entire system was implemented in a Jupyter Notebook environment to allow for iterative model tuning and visualization of training progress.

IV. RESULTS OF DNN VULNERABILITY DETECTION MODEL TRAINING

This section presents the results of the DNN vulnerability detection training. The training process considering each of the individual metrics and evaluate them simultaneously. The Figure 6 presents the comparative loss of the models during the training process. The loss is a measure of cross entropy which is the deviation in correctly predicting true from actual values.

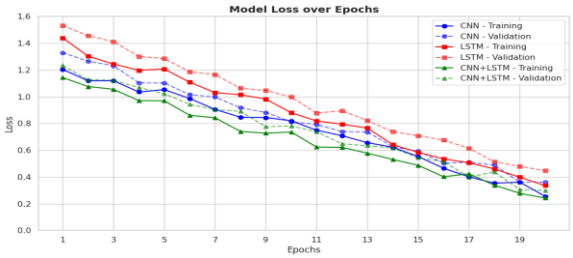

Figure 5: Loss result of the DNN vulnerability detection model in health care -IoT

Figure 5presents the training and validation loss values for three different deep learning architectures: CNN, LSTM, and a hybrid (CNN+LSTM) model. The results highlight the effectiveness of each model in identifying vulnerabilities within healthcare IoT environments. The CNN-based vulnerability detection model recorded a training loss of 0.4622 and a validation loss of 0.5323.The LSTM-based vulnerability detection model, on the other hand, performed the worst among the three models, with a training loss of 0.6310 and a validation loss of 0.6831. LSTM networks are generally suited for sequential data, making them highly effective for time-series anomaly detection.The (CNN and LSTM) hybrid model achieved the best performance, with a training loss of 0.2330 and a validation loss of 0.3853. The significant reduction in loss compared to the standalone CNN and LSTM models suggests that combining CNN's feature extraction strengths with LSTM's ability to capture sequential patterns resulted in a more robust vulnerability detection system.

Accuracy is a crucial performance metric that indicates how well a model correctly identifies vulnerabilities in a system. Higher accuracy values signify that the model is making more correct predictions, which is essential for securing healthcare IoT environments. Figure 6 presents the training and validation accuracy of the deep learning modelsused for detecting vulnerabilities in healthcare IoT systems.

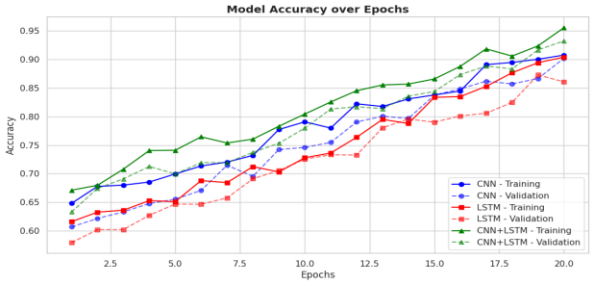

Figure 6: Accuracy of the DNN vulnerability models

These results provide insight into the models' effectiveness across different security layers, including device-level threats (perception layer), network security risks (network layer), and data protection challenges (application layer). The CNN-based vulnerability detection model recorded a training accuracy of 0.90 and a validation accuracy of 0.92. This suggests that the CNN model was effective at identifying vulnerabilities using spatial feature extraction. The LSTM-based model achieved a training accuracy of 0.91 but a lower validation accuracy of 0.86. While LSTM networks are highly effective for sequential data, the drop in validation accuracy suggests that the model may be struggling with generalization. The CNN+LSTM hybrid model outperformed both individual models, achieving a training accuracy of 0.97 and a validation accuracy of 0.93. The small difference between training and validation accuracy indicates that the model is well-regularized and less prone to overfitting, making it a more reliable solution for detecting threats across different IoT layers.

Precision is a critical metric in vulnerability detection as it measures the proportion of correctly identified vulnerabilities out of all instances that the model predicted as vulnerable. A higher precision value indicates that the model reduces false positives, ensuring that only genuine security threats are flagged, which is essential in healthcare IoT systems where false alarms can disrupt critical medical operations. Figure 7 presents the precision scores for the deep learning modelsin managing vulnerabilities across different IoT layers:

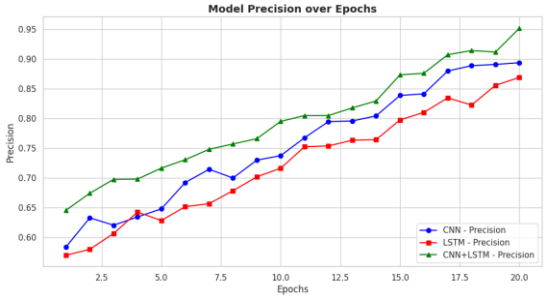

Figure 7: Precision of the DNN vulnerability management model

From the Figure 7, the CNN Model recorded a precision of 0.90, demonstrating that it effectively identifies vulnerabilities while maintaining a relatively low rate of false positives. However, their performance may be limited when dealing with evolving or sequential cyber threats. LSTM Model reported a lower precision of 0.86, suggesting that while it captures sequential relationships well, it may struggle with differentiating between actual threats and benign activities.(CNN+LSTM) Hybrid Model achieved the highest precision of 0.95, indicating its superior capability in distinguishing actual vulnerabilities from false alarms. In healthcare IoT, missing vulnerability could have severe consequences, such as compromised patient data, unauthorized access to medical devices, or system failures in life-critical applications. Figure 8 presents the recall scores for the three deep learning models used for vulnerability management:

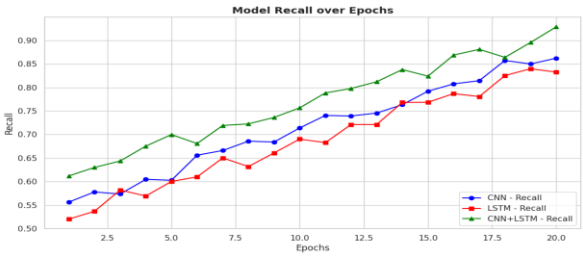

Figure 8: Recall result of the DNN vulnerability model

In the Figure 8, the CNN Model recorded a recall of 0.85, indicating that it correctly detected 85% of all actual vulnerabilities. LSTM Model achieved a recall of 0.83, slightly lower than the CNN model. (CNN+LSTM) Hybrid Model achieved the highest recall of 0.94, demonstrating its superior ability to detect the majority of actual vulnerabilities.

The F1-score is a critical metric for evaluating the effectiveness of a vulnerability detection model in healthcare IoT systems. It is the harmonic mean of precision and recall, balancing both false positives and false negatives. A higher F1-score signifies a model's ability to detect real security threats while minimizing false alarms, which is essential for healthcare IoT security, where false positives may lead to unnecessary system shutdowns and false negatives can expose patient data to cyber risks. Figure 9presents the F1-score of the models.

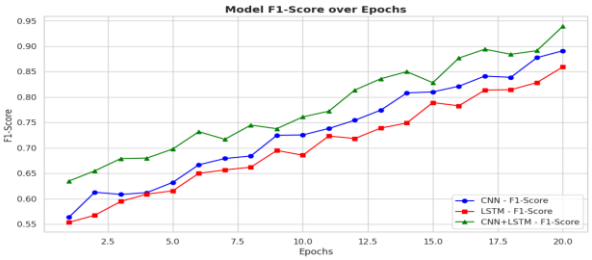

Figure 9: F1-Score of the DNN vulnerability management model

The F1-score is a harmonic mean of precision and recall, making it a crucial metric for evaluating the overall effectiveness of the deep learning models in vulnerability detection. It provides a balanced measure of how well a model correctly identifies vulnerabilities while minimizing false positives and false negatives. The CNN model achieved an F1score of 0.89%, indicating a strong balance between precision and recall. The CNN+LSTM hybrid model achieved the highest F1-score of 0.94, indicating that it is the most effective at correctly identifying vulnerabilities while maintaining a low rate of misclassification.

In the context of healthcare IoT security, where real-time detection and response are crucial, the F1-score directly affects system reliability. Higher F1-score (CNN+LSTM at 0.94) means fewer false positives and negatives, ensuring that critical vulnerabilities are detected accurately without raising too many false alarms. This is essential in healthcare, where a security breach can compromise patient data and system integrity. CNN's F1-score (0.89) suggests it performs well but may miss vulnerabilities caused by time-based anomalies, such as slow-developing attacks or delayed system responses. LSTM's F1-score (0.85) highlights its weakness in distinguishing between normal and anomalous behaviour, making it prone to false alerts, which can overwhelm security teams with unnecessary notifications. Since healthcare IoT systems handle sensitive patient data and operate in real-time, the model with the highest F1-score (CNN+LSTM) would be the most suitable for deployment, as it ensures accurate and timely threat detection while reducing operational inefficiencies.

CONCLUSION

This study presented the development of an efficient real-time vulnerability management model for cyber-physical systems. The work followed a structured approach, beginning with a technical investigation of cyber-physical systems to identify potential weaknesses. The model presented in this study provided a structural representation of security risks within the system and was further tested using real-world datasets. The analysis included evaluating different software and hardware attack vectors, enabling us to identify patterns of vulnerabilities across different system layers. To enhance security, we developed a new data model that encapsulates vulnerabilities across multiple layers of the cyber-physical system. This dataset was structured to include real-time attack scenarios and anomalies, ensuring a comprehensive representation of threats. The newly developed dataset was instrumental in training machine learning models and improving the accuracy of vulnerability detection.

A deep neural network algorithm was proposed for training the real-time vulnerability management model. This hybrid CNN+LSTM model was designed to leverage spatial and sequential learning capabilities for enhanced detection of security threats. To enhance vulnerability detection, we developed and characterized a cyber-physical system model using deep learning techniques. Three models were trained and evaluated: CNN, LSTM, and CNN+LSTM. Performance evaluation showed that the CNN+LSTM model outperformed both CNN and LSTM models, achieving a training accuracy of 0.97 and validation accuracy of 0.93, compared to CNN (0.90, 0.92) and LSTM (0.91, 0.86). The CNN+LSTM model also had the best F1-score

(0.94), precision (0.95), and recall (0.94), proving to be the most effective in identifying and managing vulnerabilities.

REFERENCES

[1] Abdelrahman, M., Nguyen, T. L., Kharchouf, I., & Mohammed, O. (2023). A hybrid physical co-simulation smart grid testbed for testing and impact analysis of cyber-attacks on power systems: Framework and attack scenarios. *Energies, 16*, 7771. https://doi.org/10.3390/en16197771

[2] Ashraf, I., Narra, M., Umer, M., Majeed, R., Sadiq, S., Javaid, F., & Rasool, N. (2022). A deep learning-based smart framework for cyber-physical and satellite system security threats detection. *Electronics, 11*(4), 667. https://doi.org/10.3390/electronics11040667

[3] Bernieri, G., Conti, M., & Pascucci, F. (2018). A novel architecture for cyber-physical security in industrial control networks. In *2018 IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI)* (pp. 1–6). IEEE. https://doi.org/10.1109/RTSI.2018.8548458

[4] Iqbal, R., Doctor, F., More, B., Mahmud, S., & Yousuf, U. (2020). Big data analytics and computational intelligence for cyber-physical systems: Recent trends and state of the art applications. *Future Generation Computer Systems, 105*, 766–778. https://doi.org/10.1016/j.future.2017.10.021

[5] Khalid, A. R., Owoh, N., Uthmani, O., Ashawa, M., Osamor, J., &Adejoh, J. (2024). Enhancing CCF detection: An ensemble ML approach. *Big Data and Cognitive Computing, 8*(1), 6. https://doi.org/10.3390/bdcc8010006

[6] Khan, S., Luo, F., Zhang, Z., Rahim, M. A., Ahmad, M., & Wu, K. (2022). Survey on issues and recent advances in vehicular public-key infrastructure (VPKI). *IEEE Communications Surveys & Tutorials, 24*, 1574–1601. https://doi.org/10.1109/COMST.2022.3154048

[7] Khazraei, A., Spencer, H., &Gao, Q. (2022). Learning-based vulnerability analysis of cyber-physical systems. In *2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)*. https://doi.org/10.1145/3517441.3526185

[8] Knowles, W., Prince, D., Hutchison, D., Disso, J., & Jones, K. (2015). A survey of cybersecurity management in industrial control systems. *International Journal of Critical Infrastructure Protection, 9*, 52–80. https://doi.org/10.1016/j.ijcip.2015.02.002

[9] Markakis, E., Nikoloudakis, Y., Pallis, E., & Manso, M. (2019). Security assessment as a service cross-layered system for the adoption of digital, personalized, and trusted healthcare. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)* (pp. 91–94). https://doi.org/10.1109/WF-IoT.2019.8767192

[10] Northern, B., Burks, T., Hatcher, M., Rogers, M., &Ulybyshev, D. (2021). VERCASM-CPS: Vulnerability analysis and cyber risk assessment for cyber-physical systems. *Information, 12*(10), 408. https://doi.org/10.3390/info12100408

[11] Parades, C. M., Martínez Castro, D., González Potes, A., Rey Piedrahita, A., & Ibarra Junquera, V. (2024). Design procedure for real-time cyber–physical systems tolerant to cyberattacks. *Symmetry, 16*(6), 684. https://doi.org/10.3390/sym16060684

[12] Segovia-Ferreira, M., Rubio Hernan, J., Cavalli, A., &Garcia-Alfaro, J. (2023). Cyber-resilience approaches for cyber-physical systems. *arXiv*. https://arxiv.org/abs/2302.05402

[13] Umer, M., Sadiq, S., Karamti, H., Alhebshi, R. M., Alnowaiser, K., Eshmawi, A. A., Song, H., & Ashraf, I. (2022). Deep learning-based intrusion detection methods in cyber-physical systems: Challenges and future trends. *Electronics, 11*(20), 3326. https://doi.org/10.3390/electronics11203326

[14] Vyas, S., & Bhargava, D. (2021). Cyber-physical systems for healthcare. In *Smart Health Systems* (pp. [chapter pages, if available]). Springer, Singapore. https://doi.org/10.1007/978-981-16-4201-2_7

[15] Wang, Y., Zhao, Z., Qi, B., Cheng, Y., Tang, K., & Li, B. (2024). Vulnerability analysis of an electric vehicle fleet for car-sharing service under cyber attacks. *Sustainable Energy, Grids*

and Networks, 37, 101207. https://doi.org/10.1016/j.segan.2023.101207

[16] Xu, J., Xue, K., Li, S., Tian, H., Hong, J., Hong, P., & Yu, N. (2019). Healthchain: A blockchain-based privacy preserving scheme for large-scale health data. *IEEE Internet of Things Journal, 6*(5), 8770–8781. https://doi.org/10.1109/JIOT.2019.2923525

[17] Yuan, S., Yang, M., & Reniers, G. (2024). Integrated process safety and process security risk assessment of industrial cyber-physical systems in chemical plants. *Computers in Industry, 155*, 104056. https://doi.org/10.1016/j.compind.2023.104056

[18] Zhou, F. Y., Jin, L. P., & Dong, J. (2017). Review of convolutional neural network. *Chinese Journal of Computers, 40*, 1229–1251.