IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 3, Issue 3, June 2023
Impact Factor: 7.301

Deep Learning-Based Fault Prediction Models for Enhanced Network Security Monitoring

Mahathi Kari
Independent Researcher
mahathikari2026@gmail.com

Abstract: *The challenge related to the security of the network operations and their reliability has gained a particular significance in the era of growing Internet of Things (IoT) ecosystems in the impact of the growing risk of cyberattacks and the system functioning collapse. The present paper describes a Deep Neural Network (DNN)-based fault prediction system that serves to complement network security monitoring by effectively detecting faults in the work of IoT network traffic. The IoT-23 dataset, a realistic and differentiated test, is used in the article. It contains both benign and malicious traffic samples. Several data preparation procedures, including cleaning, normalization, label encoding, and feature extraction, were carried out in order to optimize the model's performance. The DNN model was estimated and evaluated using common measures such as accuracy (acc), precision (pre), recall (rec), and F1-score (F1) which are calculated based on the confusion matrix. It has been experimentally verified that the proposed DNN performs better than Support Vector Machine (SVM), Naive Bayes (NB), and AdaBoost (ADA) with acc of 98.69%, 98% pre- and post-recall, and high F1. These results illustrate the power and magnitude of deep learning (DL) algorithms in fault prediction to reduce the number of false alarms, increase the detection capability, and better monitoring of the security of IoT networks.*

Keywords: Cybersecurity, Machine Learning, Network Security, Internet of Things, Fault, Machine Learning

I. INTRODUCTION

The development of optical networks to meet the growing needs of high-speed and high-capacity communication has transformed automation to become one of the most significant factors in determining network performance and scalability. The optical network automation enables operators to realize a more effective utilization of resources, resources are optimized in routing and spectral assignment and automation of failure detection and recovery [1]. Optical network automation is able to automate the work of operators, enabling them to operate more complex networks at a reduced operating cost since it also allows them to make real-time decisions and limit human intervention in the process.

The modern world is more than ever before depending on interconnected networks that make a wide sector of its infrastructure vulnerable to an enormous number of potential threats [2][3]. The field of network security is a complex topic that deals with applying intrusion detection systems, encryption protocols, firewalls, and other powerful security measures. Technology also enhances the number of tricks employed by malicious actors, thus, the network security strategies are supposed to be innovative and in a constant state of flux [4][5][6]. The propagation and developed nature of cyber threats have rendered network security a significant aspect of the modern digital infrastructure. Along with the increased reliance of organizations and individuals on interconnected systems integrity, confidentiality, and availability of data have become a most prominent concern [7][8]. Conventional security systems of the network infrastructure, which included firewalls, intrusion detection systems (IDS) and antivirus programs, were enough to guard against the known threats in the past[9][10][11]. However, the sheer speed at which attack vectors are being developed and the introduction of new and advanced forms of assaults, as advanced persistent threats (APTs) and zero-day attacks, have revealed a big weakness in these traditional methods. Data-based solutions can be used to assist in the management of faults [12][13][14]. Production lines need to have effective fault management and quick mistake correction in order to

DOI: 10.48175/IJARSCT-11600I

achieve this goal. These are the data that the ML algorithms that provide fault control utilize to carry out their functions. To increase network security and track traffic, the SDN controller may be equipped with a variety of ML/DL techniques. In recent years, the use of ML/DL techniques in network security has increased with the introduction of graphics processor units (GPUs) [15][16]. Both ML and DL are quite effective in anticipating any suspicious or malevolent network traffic activity, since they are capable of extracting and learning new features of network traffic [17]. The ML-based NIDS highly relies on the network traffic's learned features, while the DL-based NIDS may learn the intricate features straight from the raw data without using learnt features.

A. Motivation and Contribution

The rationale of this study is the increasing security dilemma due to the intensive proliferation of IoT devices, which creates enormous amounts of network flows that can be susceptible to various cyberattacks. Classical machine learning (ML) algorithms typically struggle to model the intricate trends in high-dimensional data, resulting in low accuracy and reliability in fault detection. In order to overcome these drawbacks, there is a high demand for advanced DL techniques, which can be used to elicit meaningful features, minimize false predictions, and guarantee strong performance. Using the IoT-23 dataset and building upon it a DNN model, this paper tries to develop a more accurate and efficient defect prediction algorithm that improves the security monitoring of a network and makes the IoT environment safer. This research offers several key contributions as listed below:

- Utilization of the Internet of Things (IoT)-23 dataset, which has both malicious and non-malicious captures, to ensure realistic and diverse evaluation.
- Comprehensive data pre-processing, including cleaning, noise removal, label encoding, and min–max normalization, to improve data quality and model performance.
- Application of feature extraction techniques to reduce complexity and increase the categorization process.
- Providing a scalable and reliable solution to improve IoT security monitoring and network failure prediction.
- Development of a DNN model specifically designed for fault prediction in IoT-based network environments.
- In-depth analysis of the suggested model with respect to critical performance indicators like rec, acc, pre, and F1.

B. Organization of the Paper

The structure of this paper is as follows: Section II provides a review of related work on fault prediction for enhanced network security monitoring, Section III details the dataset, procedures for preparation, and the application of the model, Section IV offers a comparative examination of the experimental findings, and Section V the paper by reviewing its main points and suggesting avenues for further investigation.

II. LITERATURE REVIEW

A thorough review and analysis of the significant development of this study was guided and strengthened by previous research on fault prediction.

Alqahtani and Clark (2022) identify and forestall, as their features are more complex than those of conventional assaults. When it comes to SDNs, there hasn't been much study on APT detection. They test the efficacy of ML algorithms in detecting APT scanning operations within SDN. For the suggested detection model, they use the XGBoost classifier to attain a minimum of 97.8% in F1-measures, Acc, Rec, and Pre with just five characteristics. Experiment datasets across a range of network sizes are created and made publicly available for free usage [18].

Uppal et al. (2022) The suggested fault prediction model was evaluated using DT, KNN, Gaussian NB, and RF methodologies, nevertheless, RF performed the best on the presented dataset. The outcomes demonstrated the effectiveness of ML approaches applied to IoT-based sensors for tracking this automation procedure in hospitals. With a maximum accuracy of 94.25%, (RF) was shown to be the most efficient. The suggested model could assist the user in deciding which option is best and managing unforeseen losses brought on by errors in the automation process [19].

Kodali and Muntean (2021) A DNN that was trained using 28 features from the NSL-KDD dataset makes up the suggested system. The DNN has shown remarkable testing performance results, achieving 81%, 96%, 70%, and 81%

DOI: 10.48175/IJARSCT-11600I

for acc, pre, rec, and f1, respectively. An extensive technical explanation of the system's implementation and operation is included in this study [20].

Shahzadi et al. (2020) To get around these problems, there has to be an independent system that can identify and describe any unusual behaviour in network data. ML-based characterization techniques like SVM, K-Nearest Neighbours (KNN), Logistic Regression (LR), and Isolation Forest (IF) are used to optimize the security management and stability of the SDN-NFV system. The dataset contains only four attack types: Smurf Flood, HTTP Flood, UDP Flood, and SiDDoS Flood. The open-source Python ML libraries Scikit-learn, NumPy, SciPy, Matplotlib, and the mine package are some useful tools used in Python simulations. Regardless of whether the network traffic is typical or unusual in the SDN-NFV environment, the encouraging classification findings demonstrate that the overall accuracy for SVM, LR, KNN, and IF classifiers in the traffic analysis ranges from 87% to 95% [21].

Lee et al. (2019) focus on distinguishing real positive alarms from false positive signals, which helps security experts respond swiftly to online threats. The authors of this research employed two real-world datasets and two benchmark datasets (NSLKDD and CICIDS2017) for every experiment. Examined the performance of the five traditional ML techniques (SVM, k-NN, RF, NB, and DT) in comparison to other methods. As a consequence, the study's experimental findings demonstrate that suggested techniques may be used as learning-based models for network intrusion detection and that, despite their practical use, they perform better than traditional ML techniques [22].

Al-Garadi et al. (2018) describe in detail the latest developments in DL techniques and ML techniques that may be applied to the development of improved security measures for IoT systems. A variety of potential IoT system attack surfaces are examined, along with the potential dangers associated with each surface. Risks to IoT security that are linked to either new or existing risks are shown. After that, provide a detailed analysis of ML/DL techniques for IoT security, including their benefits, drawbacks, and potential applications. It goes over the potential and difficulties of integrating ML/DL with IoT security. Future study avenues may be suggested by these opportunities and difficulties [23].

Table I, Although ML and DL techniques have shown effectiveness in fault prediction and intrusion detection across SDN, NFV, and IoT environments, gaps remain in detecting advanced persistent threats, handling multi-attack scenarios, and ensuring real-time adaptability in large-scale, dynamic networks. Most studies focus on accuracy over practical deployment, and few integrate solutions across diverse domains. There is a need for robust, efficient, and scalable models capable of real-time detection in complex networked systems

Table 1: Recent Studies on Fault Prediction for Enhanced Network Security

Author	Proposed Work	Results	Key Findings	Limitations & Future Work
Alqahtani and Clark (2022)	Designed a machine learning-based APT scanning detection model within SDN using XGBoost classifier.	Achieved ≥97.8% across Accuracy, Recall, Precision, and F1-measure using 5 optimized features.	Demonstrated effective ML detection of stealthy APT scanning activities in SDN.	Further exploration needed for adaptive detection of evolving APT behaviors and deployment in real-world SDN infrastructures.
Uppal et al., (2022)	Fault prediction model using Decision Tree, KNN, Gaussian Naive Bayes, and Random Forest on IoT-based hospital automation sensors	Random Forest achieved highest accuracy: 94.25%	ML techniques efficiently monitor hospital automation processes; Random Forest performs best among evaluated models	Limited to IoT sensor data; future work could explore real-time adaptive fault prediction and integration with other hospital systems
Kodali & Muntean, (2021)	Deep Neural Network (DNN) trained on 28 features of NSL-KDD dataset	Accuracy: 81%, Precision: 96%, Recall: 70%, F1-	DNN shows strong testing performance and effective feature-	The dataset limited to NSL-KDD; future work could include larger datasets, real-

		score: 81%	based learning	world deployment, and multi-class attack detection
Shahzadi et al., (2020)	Autonomous anomaly detection in SDN-NFV using SVM, KNN, LR, and Isolation Forest for four attack types (HTTP Flood, UDP Flood, Smurf Flood, SiDDoS)	Accuracy between 87% - 95%	ML-based characterization effectively identifies anomalous network traffic and optimizes SDN-NFV security	Only four attack types considered; future research could extend to multi-attack scenarios and real-time adaptive systems
Lee et al., (2019)	Network intrusion detection focusing on discriminating true and false positive alerts using SVM, KNN, RF, NB, and DT on NSL-KDD, CICIDS2017, and real-world datasets	Outperforms conventional ML methods in real-world datasets	Learning-based models improve alert accuracy, helping analysts respond faster to threats	Future work could focus on large-scale, heterogeneous networks and improving real-time detection capabilities
Al-Garadi et al., (2018)	An examination of ML and DL techniques for Internet of Things safety	Examined the potential, benefits, and drawbacks of each approach.	Provides thorough explanations of ML/DL for IoT security as well as possible avenues for further study	Survey-based; future work could involve practical implementation of hybrid ML/DL frameworks for IoT security

III. RESEARCH METHODOLOGY

The aim of this study is to develop an effective and robust fault prediction using ML and DL methods. Its emphasis is on enhancing real-time monitoring and security in IoT, SDN, and NFV environments.

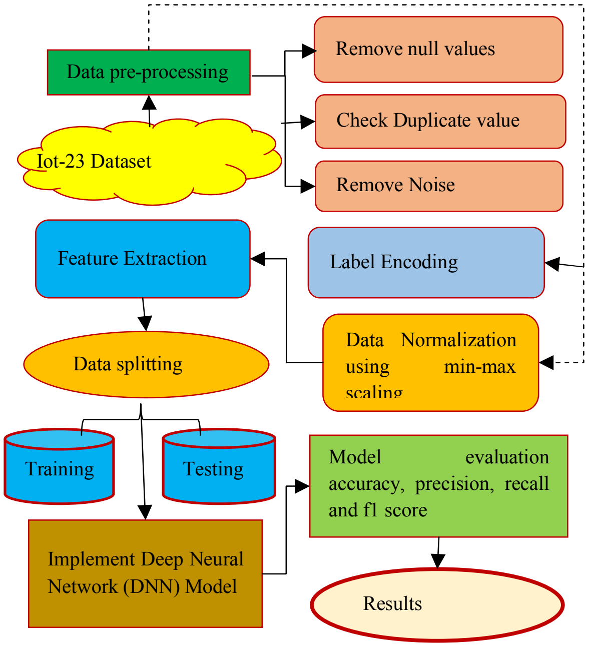

Fig. 1. Proposed Flowchart for Fault Prediction for Enhanced Network Security Monitoring

The methodology of this study involves a systematic process beginning with data gathering and analysis using the IoT-23 dataset. Pre-processing was then carried out to prepare the dataset, including the following steps are illustrated in Figure 1 below:

The following section presents an exhaustive breakdown of the suggested procedure:

A. Data Gathering and Analysis

A new dataset, IoT-23, contains information on network traffic generated by IoT devices. Twenty instances of malware detected by different IoT devices and three instances of benign abnormalities make up the dataset. The data was gathered between 2018 and 2019 in collaboration with the Czech Technical University in Prague. Data visualizations such as bar plots and heatmaps were used to examine attack distribution, feature correlations etc., are given below:

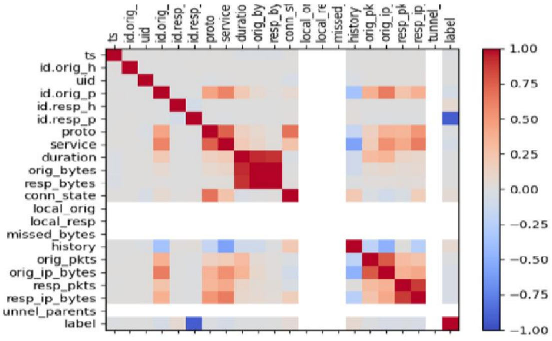

Fig. 2. Correlation Matrix of Data

Figure 2 represents a correlation heatmap of different network traffic features, where the colour scale goes from -1 (significantly negative correlation, represented in blue) to +1 (strongly positive correlation, indicated in red). Dark red squares on a diagonal line show that all features are perfectly correlated with each other. Several sets of characteristics, shown by deeper red blocks, show substantial positive correlations with each other, suggesting that they change together. These sets include orig_bytes, resp_bytes, orig_ip_bytes, resp_ip_bytes, and resp_pkts. Conversely, some features like label and history show weak or negative correlations with other variables, indicated by patches of blue.

B. Data Pre-processing

The IoT-23 dataset was utilized for data preparation, which involved concatenation, data cleaning, and feature engineering. The pre-processing steps included handling missing values, detecting and removing duplicates, eliminating noise, followed by data labelling and normalization. The most significant pre-processing steps are described in the following way:

- **Remove null values:** Every column is examined for the existence of null or NaN values and suitable solutions for imputation or removal are decided upon as part of this procedure.
- **Check Duplicate value:** To provide trustworthy analytical results and equitable models, duplicates must be found and eliminated.

Remove Noise: To remove noise is the process of eliminating unwanted, disruptive signals or artifacts from a primary signal, such as audio, image, or data, to improve its clarity, quality, and usefulness

C. Label Encoding

In label encoding, numerical labels are given to each category in order to convert categorical input into numerical data. One of the sclera packages, Label Encoder, might be applied to convert numerical data from categories into the appropriate labels.

D. Min-Max Normalization

The data were normalized using the min-max approach, which limits values to a range of 0 to 1. The goal of doing this

DOI: 10.48175/IJARSCT-11600I

was to lower the impact of outliers and optimize the performance of the classifiers. The following formula was used to execute the normalization process Equation (1):

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (1)$$

Where X stands for the feature's initial value, X' for its normalized X_{min} for its minimum value, and X_{max}Or its maximum value.

E. Feature Extraction

Feature extraction is an ML step of summarizing large and complex data into a smaller and easier-to-manage set of features that conveys the most important information. It improves model performance and lowers computing costs. Enhancing the accuracy of the model, avoiding overfitting, and making the training and prediction process faster are all the objectives of this crucial pre-processing phase that implies converting raw data into a form that is more formatted and easily available with a focus on relevant characteristics. This technique reduces data complexity and dimensionality, leading to faster, more efficient, and often more accurate models.

F. Classification with Deep Neural Network (DNN) Model

A well-known DL algorithm is the DNN. Each layer of a DNN network the input, hidden, and output layers—is fully coupled to the others. Every neuron communicates only with its immediate neighbours in the same layer, and not with any neurons in lower or higher levels[24]. An activation function enhances the impact of network learning by acting on the result following every network layer. Thus, DNN may alternatively be seen as a network of many little perceptron's working together. For the forward propagation computation of the ith layer, for instance, the Equation is (2):

$$x_{i+1} = \sigma(\sum w_i x_i + b) \qquad (2)$$

Where x stands for the input value, q for the weight coefficient matrices, and b for the bias vector, as an activation function, ReLU is commonly employed in multi-class networks; the Equation for this is (3):

$$\sigma(x) = max(0, x) \qquad (3)$$

The network structure is determined by the loss function, which optimizes the network's backpropagation and evaluates training sample output loss. The loss function in classification issues is often cross-entropy, its Equation is as follows (4):

$$C = -\frac{1}{N}\sum_x \sum_{i=1}^{M} (y_i log p_i) \qquad (4)$$

This is where N is the input data set count, M is the number of categories, yi is the probability of predicting into category i, and pi is the probability that category i corresponds to the real category. After activation using ReLU, the Adam optimizer was used on the DNN.

G. Evaluation Metrics

Several performance criteria were employed to assess the proposed design's efficacy. First of all, a confusion matrix was created to generalize the results of classification, which included the quantity of accurate and inaccurate forecasts across the different classifications. The critical values obtained included TP, FP, TN and FN. Details on how these parameters were used to compute key performance metrics, including as F1, rec, acc, and pre, are provided below:

- **True Positive (TP):** The frequency with which the prediction came true as *fault* when they are actually *fault* (i.e., correct identification of faulty network traffic).
- **True Negative (TN):** The proportion of cases that were accurately predicted as *non-fault* when they are actually *non-fault* (i.e., correct identification of normal or benign network traffic).
- **False Positive (FP):** The total number of cases when *fault* was wrongly projected when there was none.
- **False Negative (FN):** The total number of instances in which *non-fault* was mistakenly predicted as *fault* and vice versa.

Accuracy

The proportion of the dataset's occurrences that trained model accurately predicted, expressed as based on overall number of occurrences, is Equation (5)-

$$Accuracy = \frac{\text{TP+TN}}{\text{TP+Fp+TN+FN}} \qquad (5)$$

Precision

The ratio of successfully predicted positive instances to all positive examples is a measure of a model's prediction accuracy. Accuracy signifies how good the classifier is in predicting the positive classes and is expressed as Equation (6)-

$$Precision = \frac{\text{TP}}{\text{TP+FP}} \qquad (6)$$

Recall

This statistic shows the percentage of anticipated positive results that actually occurred out of all anticipated negative occurrences. It expressed mathematically as Equation (7)-

$$Recall = \frac{\text{TP}}{TP+FN} \qquad (7)$$

F1 score

It combines the harmonic mean of memory and accuracy to determine the balance between recall and precision. Its range is [0,1]. Mathematically, it is given as Equation (8)-

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \qquad (8)$$

Using these evaluation metrics, the next section examines and discusses the computed results

IV. RESULTS AND DISCUSSION

The results and evaluation of the tests conducted using Python 3.9 and Scikit-learn on a system with an Intel Core i7 CPU, 16 GB RAM, and 512 GB SSD, NumPy, SciPy, Matplotlib, and TensorFlow/Keras for ML and DL model implementation are presented in this section. The classification performance of the proposed DNN model for predicting failures on the IoT-23 dataset is shown in Table II. The model achieved 98.69% accuracy, indicating its strong ability to correctly detect faults and non-faults. Moreover, the DNN model successfully balances false positives and false negatives, delivering strong and reliable defect detection performance, as evidenced by its 98% accuracy, recall, and F1-scores. These findings confirm the strength and transfer capability of DNN configuration in analyzing IoT network data to diagnose faults beforehand.

Table 2: Classification results of DNN model, Fault Prediction using IoT-23 dataset

Matrix	Deep Neural Network (DNN) Model
Accuracy	98.69
Precision	98
Recall	98
F1-score	98

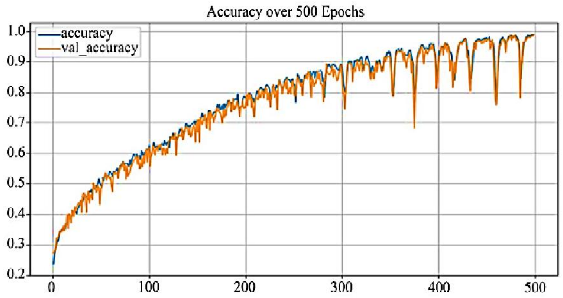

Fig. 3. Loss Curves for the CNN Model

Figure 3 presents the suggested model's accuracy following 500 training and validation cycles. Both the accuracy and the validation accuracy show a steady increase starting at a point below 0.2 and gradually increasing with the increase in the number of epochs. The model is also very accurate with the acc values of the model at the end of the training period always being above 0.9 although with slight fluctuations in the validation curve because of natural variations during testing.

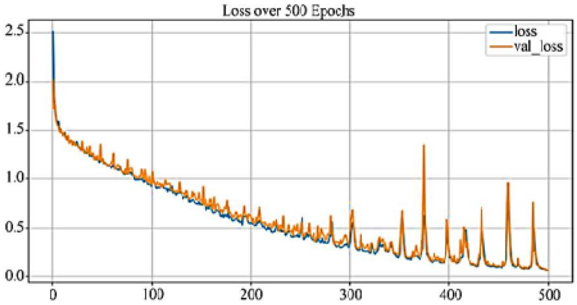

Fig. 4. Loss Curves for the CNN Model

Figure 4 shows the model's training and validation losses across 500 epochs. Both loss values start off high, above 2.5, but they drop quickly in the early epochs, which means learning is going well. The loss keeps going down as training goes on, eventually levelling out at less than half after around 300 epochs. While the training loss remains smooth and consistently decreases, the validation loss exhibits fluctuations, with occasional spikes throughout the later epochs.

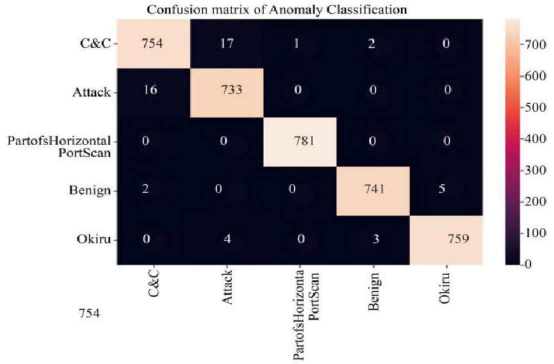

Fig. 5. Confusion Matrix for DNN Model

In Figure 5, the findings of a model that looks for anomalies in five different classes are shown in this confusion matrix: Command and Control, Attack, Part of Horizontal-PortScan, Benign, and Okiru. The diagonal values, such as 754 for C&C and 733 for Attack, represent a high degree of accuracy, as shown by the quantity of cases accurately recognized for every class. With few exceptions, the model successfully differentiates between these five categories of network traffic, as shown in the matrix.

A. Comparative Analysis

An assessment of the relative efficacy of Table III displays many ML models for predicting network security breaches, including SVM, NB, ADA, and the proposed DNN. At 98% pre, 98% rec, 98% F1, and 98.69% acc, the DNN model demonstrated the best predictive capability and dependability of all the models evaluated. Traditional models, including SVM and NB, had moderate results, whereas ADA achieved rather better results with an accuracy of 87%.

Table 3: Comparison of Different Models for Fault Prediction for Network Security

Model	Accuracy	Precision	Recall	F1-score
SVM[25]	74	70	74	70
NB[26]	78.8	78.8	78.8	78.8
ADA[27]	87	86	87	83
DNN	98.69	98	98	98

The advantage of the proposed DNN model is its incredibly high accuracy of 98.69%, which is better than traditional ML methods and provides a high degree of dependability in fault prediction. The model enables the reduction of false alarms, fault detection, and balanced performance through the model's efficacy in improving network security, Using the IoT-23 dataset is responsible for its consistently good pre, rec, and F1, acc.

B. Limitations and Future Work

Although the DNN model proposed has a high performance in predicting failures in the context of monitoring the security of IoT networks, some limitations can be identified. The current plan mostly concerns a set of fixed data (IoT-23), which ignores the constantly changing character of cybercrimes and real-time network traffic. Moreover, the interpretability of the model is low, and one may find it difficult to rationalize the decision-making process to the network administrators in emergency cases. Further development in the future will be aimed at making the model more resilient to hostile attacks, making it more interpretable with the help of XAI methods, and improving its ability to adapt to streaming and real-time inputs. In addition, incorporating hybrid DL systems, like CNN LSTM and attention-based models, and federated learning systems to train on distributed and privacy-sensitive training environments, will improve scalability, generalization and security in varied and large-scale IoT systems.

V. CONCLUSION AND FUTURE STUDY

In the fast-changing context of digital communications, network security has become one of the most critical concerns, especially considering the complexity of the modern network architecture and the accelerating astronomical growth in the quantity of connected IoT devices. Traditional ML methods are not always effective in capturing complex nonlinear trends in large and high-dimensional IoT traffic data because of their limitations in previous settings. This study developed and implemented a DNN model using the IoT-23 dataset to forecast network security monitoring failure. The suggested DNN outperformed other conventional models like SVM (74%), NB (78.8%), and AdaBoost (87%) in terms of pre, rec, and F1 of 98%, achieving an exceptional acc of 98.69%. The model is also capable of reducing false positives and negatives due to balanced recall and good accuracy, which guarantees scalable and real-time detection of network security problems. The results validate the robustness of the DNN architecture to act on complex data in the IoT network and provide evidence of its possible applicability in the implementation of intelligent and automated systems to monitor the network. The study shows that DL is superior to the common ML models in network fault prediction and provides a framework that can be scaled to support the use of proactive security management in the actual IoT environment.

REFERENCES

[1] R. Gu, Z. Yang, and Y. Ji, “Machine learning for intelligent optical networks: A comprehensive survey,” 2020. doi: 10.1016/j.jnca.2020.102576.

[2] H. Ruan, B. Dorneanu, H. Arellano-Garcia, P. Xiao, and L. Zhang, “Deep Learning-Based Fault Prediction in Wireless Sensor Network Embedded Cyber-Physical Systems for Industrial Processes,” *IEEE Access*, vol. 10, pp. 10867–10879, 2022, doi: 10.1109/ACCESS.2022.3144333.

[3] F. Yan, Y. Jian-Wen, and C. Lin, “Computer Network Security and Technology Research,” in *2015 Seventh International Conference on Measuring Technology and Mechatronics Automation*, IEEE, Jun. 2015, pp. 293–296. doi: 10.1109/ICMTMA.2015.77.

[4] W. Xing, “Research on Computer Network Security Problems and Countermeasures Research on Computer Network Security Problems and Countermeasures,” *J. Phys. Conf. Ser.*, 2021, doi: 10.1088/1742-6596/1992/3/032069.

[5] V. Shah, “Managing Security and Privacy in Cloud Frameworks: A Risk with Compliance Perspective for Enterprises,” *Int. J. Curr. Eng. Technol.*, vol. 12, no. 6, pp. 1–13, 2022.

[6] G. Sarraf, “Behavioral Analytics for Continuous Insider Threat Detection in Zero-Trust Architectures,” *Int. J. Res. Anal. Rev.*, vol. 8, no. 4, pp. 596–602, 2021.

[7] M. Z. Alom *et al.*, “A State-of-the-Art Survey on Deep Learning Theory and Architectures,” *Electronics*, vol. 8,

no. 3, p. 292, Mar. 2019, doi: 10.3390/electronics8030292.

[8] K. C. S. Ramakrishnan Sundaram Senthilkumar Thangavel, Suresh Bysani Venkata Naga, "Secure and Scalable Data Replication Strategies in Distributed Storage Networks," *Int. J. AI, BigData, Comput. Manag. Stud.*, vol. 2, no. 2, pp. 18–27, 2021.

[9] J. Kachhia, R. Natharani, and K. George, "Deep Learning Enhanced BCI Technology for 3D Printing," in *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, IEEE, Oct. 2020, pp. 0125–0130. doi: 10.1109/UEMCON51285.2020.9298124.

[10] V. Pal, "Bias Detection and Mitigation in Foundation AI Models: A Human-Centric Approach," *TIJER – Int. Res. J.*, vol. 8, no. 2, pp. 1–7, 2021, [Online]. Available: https://tijer.org/tijer/papers/TIJER2102002.pdf

[11] S. Garg, "Predictive Analytics and Auto Remediation using Artificial Inteligence and Machine learning in Cloud Computing Operations," *Int. J. Innov. Res. Eng. Multidiscip. Phys. Sci.*, vol. 7, no. 2, 2019.

[12] W. Gong *et al.*, "A Novel Deep Learning Method for Intelligent Fault Diagnosis of Rotating Machinery Based on Improved CNN-SVM and Multichannel Data Fusion," *Sensors*, vol. 19, no. 7, p. 1693, Apr. 2019, doi: 10.3390/s19071693.

[13] G. S. Krishna, "Research On Computer Network Security Based On The Era Of Big Data," *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 02, no. 06, pp. 1069–1072, 2020.

[14] R. K. Safia Malallah, Yasser Zalah, "An Analysis of the Advanced Encryption Standard and Threats Associated," *Researcg Gate*, pp. 1–9, 2018.

[15] A. Dawoud, S. Shahristani, and C. Raun, "Deep learning and software-defined networks: Towards secure IoT architecture," *Internet of Things*, vol. 3–4, pp. 82–89, Oct. 2018, doi: 10.1016/j.iot.2018.09.003.

[16] S. Chountasis, D. Pappas, and D. Sklavounos, "Network intrusion detection method based on matrix factorization of their time and frequency representations," *ETRI J.*, vol. 43, no. 1, pp. 152–162, Feb. 2021, doi: 10.4218/etrij.2019-0476.

[17] Y. Li, G. Huang, C. Wang, and Y. Li, "Analysis framework of network security situational awareness and comparison of implementation methods," *J. Wirel. Commun. Netw.*, pp. 1–32, 2019.

[18] A. H. Alqahtani and J. A. Clark, "Enhanced Scanning in SDN Networks and its Detection using Machine Learning," in *2022 IEEE 4th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)*, IEEE, Dec. 2022, pp. 188–197. doi: 10.1109/TPS-ISA56441.2022.00032.

[19] M. Uppal *et al.*, "Cloud-Based Fault Prediction for Real-Time Monitoring of Sensor Data in Hospital Environment Using Machine Learning," *Sustain.*, 2022, doi: 10.3390/su141811667.

[20] S. K. Kodali and C. H. Muntean, "An Investigation into Deep Learning Based Network Intrusion Detection System for IoT Systems," in *Proceedings of 2021 IEEE International Conference on Data Science and Computer Application, ICDSCA 2021*, 2021. doi: 10.1109/ICDSCA53499.2021.9650111.

[21] S. Shahzadi *et al.*, "Machine Learning Empowered Security Management and Quality of Service Provision in SDN-NFV Environment," *Comput. Mater. Contin.*, vol. 66, no. 3, pp. 2723–2749, 2021, doi: 10.32604/cmc.2021.014594.

[22] J. Lee, J. Kim, I. Kim, and K. Han, "Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles," *IEEE Access*, 2019, doi: 10.1109/ACCESS.2019.2953095.

[23] M. A. Al-garadi, A. Mohamed, A. Al-ali, X. Du, and M. Guizani, "A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security," *Res. Gate*, vol. 22, no. 3, pp. 1646–1685, 2020, doi: 10.48550/arXiv.1807.11023.

[24] A. Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: a review," *Artif. Intell. Rev.*, 2020, doi: 10.1007/s10462-019-09794-5.

[25] L. Gotsev, M. Dimitrova, B. Jekov, E. Kovatcheva, and E. Shoikova, "A cybersecurity data science demonstrator: Machine learning in IoT network security," *25th World Multi-Conference Syst. Cybern. Informatics, WMSCI 2021*, vol. 2, no. Wmsci, pp. 1–6, 2021.

[26] R. Thamaraiselvi and S. A. S. Mary, "Attack and Anomaly Detection in IoT Networks using Machine Learning," *Int. J. Comput. Sci. Mob. Comput.*, vol. 9, no. 10, pp. 95–103, Oct. 2020, doi:

10.47760/ijcsmc.2020.v09i10.012.

[27] N. A. Stoian, “Machine Learning for Anomaly Detection in IoT networks: Malware analysis on the IoT-23 Data set,” *Univ. Twente*, 2020.