A ROBUST MACHINE LEARNING APPROACH FOR INTELLIGENT RANSOMWARE DETECTION IN INDUSTRIAL CONTROL NETWORKS

Mr. Swapnil Joshi
Assistant Professor
Department of Computer Sciences and Applications
Mandsaur University, Mandsaur
swapnil.joshi@meu.edu.in

To Cite this Article
Mr. Swapnil Joshi, "A Robust Machine Learning Approach For Intelligent Ransomware Detection In Industrial Control Networks", Journal of Science Engineering Technology and Management Science, Vol. 03, Issue 05, May 2026, pp: 21-28, DOI: http://doi.org/10.64771/jsetms.2026.v03.i05.pp21-28
Submitted: 19-03-2026 Accepted: 25-04-2026 Published: 02-05-2026

***Abstract*—Malicious software known as Malware in the form of viruses, ransomware, and spyware has turned into a global epidemic, and research shows that the impact is intensifying. Numerous ways have been introduced to date to deal with these hazards. To handle this increasing problem, this paper proposes an effective Deep Neural Network (DNN) model that can be used to detect ransomware precisely. The model proves to be very effective in the separation of malicious and benign samples, with an accuracy, precision, recall, and F1-score of 99.76, and an AUC value of 0.98, which indicates the close to perfection of the classification. The findings reveal the high learning stability and generalization without overfitting, which is reinforced by the stable training and validation. Compared to the current methods, including KNN (83.9%), VGG-16 (90.5%), XGBoost (94.1%), and Logistic Regression (96%), the DNN-based model was better in its performance. On the whole, this paper highlights how deep learning can be used to reinforce cybersecurity protection and offer a scalable and intelligent method to counter ransomware attacks in the present digital environment.**

Keywords—Ransomware Detection, Industrial Control Networks (ICNs), Machine Learning (ML), Cybersecurity, Anomaly Detection, Intelligent Systems, SCADA Security.

This is an open access article under the creative commons license https://creativecommons.org/licenses/by-nc-nd/4.0/
CC BY-NC-ND 4.0

I. INTRODUCTION

The Industrial Control Networks (ICNs) have emerged as a part of the contemporary industrial processes, and it is utilized in manufacturing, energy, transportation, and critical infrastructure[1][2]. Cyber threats have become sophisticated to attack these networks that combine the physical and digital world by Supervisory Control and Data Acquisition (SCADA) systems and Programmable Logic Controllers (PLCs)[3]. Ransomware is one of the most devastating types of attack vectors in this list, where key operational information is encrypted and a ransom price is paid to decrypt it[4]. The impacts of such attacks are not limited to financial damage as they lead to long production processes[5], safety and security threats and could lead to interruption in the national critical infrastructure, further pushing the urgency of these intelligent detection and prevention systems[6][7][8]. The conventional methods of cybersecurity, which are mainly signature-based and rule-driven, have difficulty in dealing with the adaptive character of the contemporary ransomware[9][10]. Such traditional systems cannot be used to fight zero-day attacks and other ransomware no-fly zones that constantly adapt to avoid familiar defensive measures[11][12][13][14]. Hence, the trend is moving towards using smart and intelligent security systems that are able to detect irregular activities in ICNs independently[15][16].

Machine Learning (ML) has demonstrated enormous potential in this area and has a feature of being able to identify intricate patterns of actions based on the past and also differentiate between malicious activity and benign operations. Through real-time network traffic, process variables, and system-level behavior analyses, ML-based models can early identify ransomware and drastically decrease the detection latency and enhance the industrial system resilience[17][18][19]. To strengthen cybersecurity in industrial control systems, the suggested study offers a clever ransomware detection solution based on efficient machine learning models[20][21]. The framework combines the state-of-the-art data pre-processing, feature selection and classification algorithms to recognize patterns of ransomware accurately and with the least false positives. The approach is able to deliver a high detection rate, computational efficiency, and scalability, which makes it appropriate for real-time applications in industries. Finally, the proposed study can be used to strengthen industrial cybersecurity by offering a proactive, dynamic, and intelligent protection system that would help to alleviate new ransomware attacks in industrial control systems.

A. Motivation and Contribution

The fast pace of ransomware attack development and its growing complexity present a significant risk to contemporary computing systems, leading to the massive loss of money and information in the industries. Conventional machine learning approaches and signature-based traditional methods may not be generally applicable to these emerging and changing types of ransomware. This inspires the necessity of a smart, data-driven detection model that is capable of proficiently acquiring intricate behavioural patterns via large-scale data like GCRD. Using deep learning, this project enhances the accuracy of detection, the resilience to unseen attacks, and provides a stable solution to proactive ransomware prevention. This study contributes in a number of ways, as enumerated below. This research offers several key contributions as listed below:

- Proposed complete ransomware detection system based on the ransomware dataset with an equal measure of ransomware and benign samples.
- A strong data cleaning pipeline, outlier removal, z-score normalization, and label encoding are developed to improve the quality of data.
- Graph-Based Feature Selection (GFS) used to select the most important features, which reduces the dimensions of the model and increases its efficiency.
- A Deep Neural Network (DNN)-oriented detection model is created to acquire knowledge of sophisticated ransomware action patterns.
- To evaluate the model's comprehensiveness and provide a reliable evaluation of its performance, many performance measures were used, such as accuracy, precision, recall, F1-score, and ROC-AUC.

B. *Justification And Novelty*

The originality of this study is the combination of the Deep Neural Network and Graph-Based Feature Selection to ransomware behavior that is not easy to observe, and enhances the success of detection. However, as opposed to traditional machine learning models, which cannot readily detect subtle malicious patterns in the presence of nonlinear interactions among features, the proposed framework takes advantage of the hierarchical representation capability of deep learning to detect subtle malicious patterns with high accuracy. The features' relevance and the model generalization are improved by such a combination, thus the resilience to various ransomware variants is good. This methodology is justified by the fact that it offers a smart, dynamic and scalable detection engine, which renders it a big leap as compared to the current traditional and superficial learning-based cybersecurity frameworks.

C. *Organization of the Paper*

The paper is structured as follows: Section II presents related work on ransomware detection, Section III outlines dataset and pre-processing, and proposes the model, Section IV presents the experimental results and comparative analysis, and conclusion of the research is given in Section V.

II. LITERATURE REVIEW

The main research works conducted on the topic of ransomware detection in cybersecurity were reviewed and critically analyzed, and the information presented in the work was to direct the creation of the current one and enhance its strength.

Souza and Batista (2025), Seven ML classifiers are available for training and comparison: KNN, LR, RF, SVM, MLP, NB, and XGBoost. A mean classification time of 82.15 ms is provided by Random Forest, which also obtains the maximum accuracy (99.33%). Following closely are the Logistic Regression (96.80%) and K-Nearest Neighbours (97.33%). In order to promote more study in the area, the dataset is made freely accessible[22].

Kipanga and Khennou, (2025) analyzed different dataset segments' impact on ML algorithms, refining a strategy to determine the optimal dataset proportion for training. Seven ML models were tested alongside DL models. The LSTM model's ransomware detection accuracy was 99.7%. In the Malware dataset, an accuracy of 99.9% was obtained across all evaluation metrics using only 10 features selected with the Chi-square method when applied with NB, LR, ET, and SVM. Comparable results were achieved using mutual information in conjunction with RF and LR. For deep learning, the LSTM model attained an accuracy of 98.9%. In the Ransomware dataset, an accuracy of 99.9% was achieved using RF and ET with Chi-square on 500 selected features out of 1,027. PCA combined with LR resulted in an accuracy of 99.4%[23].

Polamarasetti, (2024) found that malicious computer traffic may be discovered through research into ML techniques for malware research and detection, which could lead to improved network security. NB, SVM, RF, and J48 were four algorithms used. The data showed that DT (99%), CNN (98.76%), and SVM (96.41%) were the top three classifiers in terms of detection accuracy. On a dedicated dataset, tested DT, CNN, and SVM for malware identification using a tiny FPR. The results for DT, CNN, and SVM were 2.01%, 3.97%, and 4.63%, respectively. The increasing prevalence and complexity of malicious software make these findings noteworthy[24].

Baksi, Nalka and Upadhyaya, (2023) Compare the many intrusion detection systems created with the six types listed above. The accuracy of the IDS using the Naive Bayes Classifier is 98.55%, but IDS using NLP BERT model has a maximum accuracy of 99.98%. Discuss the compromises between these methods to develop an intelligent IDS as well. The development of cyberattacks, particularly ransomware-based assaults, makes this IDS update necessary for a robust defense [25].

Aljubory and Khammas (2021) suggest three ML methods—RF, SVM, and Bayes—to identify and categorize ransomware. To expedite detection, a feature set was instantly created from raw bytes using the static analysis approach of samples. CF-NCF (Class Frequency - Non-Class Frequency) has been used to create feature vectors in order to maximize detection accuracy. The suggested method has a 98.33% detection accuracy in differentiating between ransomware and good ware files[26].

Basnet et al. (2021) evaluate the effectiveness of three DL algorithms—deep neural network (DNN), 1D convolution neural network (CNN), and long short-term memory (LSTM) RNN—to develop a novel DL-based ransomware detection framework for SCADA-controlled electric vehicle charging stations (EVCS). The three DL-based simulated frameworks showed an average accuracy (ACC) of over 97%, an average area under the curve (AUC) of over 98%, and an average F1 of less than 1.88% during 10-fold stratified cross-validation. By surpassing the SOC control thresholds, ransomware-driven distributed denial of service (DDoS) assaults frequently modify the state of charge (SOC) profile[27].

Although the current ML and DL-based ransomware detection models have high detection accuracy, most studies use limited or domain-specific datasets, and thus, their applicability to a real-world environment is limited. Also, a number of methods are computationally intensive or feature-rich and therefore cannot be deployed in real time. Thus, lightweight, scalable, and general-purpose detection frameworks that achieve high accuracy and efficiency across ransomware variants and dynamic network conditions are required.

III. Research Methodology

The approach to the research is a methodical process comprising data gathering[28], pre-processing, model building, and validation. The pre-processing of a balanced ransomware dataset included imputation, removal of duplicates and outliers, label coding, and normalization of z-score. Graph-Based Feature Selection (GFS) selected important features to enhance the acc of model. The 70:30 train-test split and Deep Neural Network (DNN) with cross-entropy loss were applied to the data. Performance parameters including acc, prec, rec, F1, and ROC curve were used to evaluate the model's reliability and ensure that the ransomware was effectively identified. The flowchart proposed is depicted in Figure. 1.

A thorough description of each step in the suggested technique is provided in the section that follows:

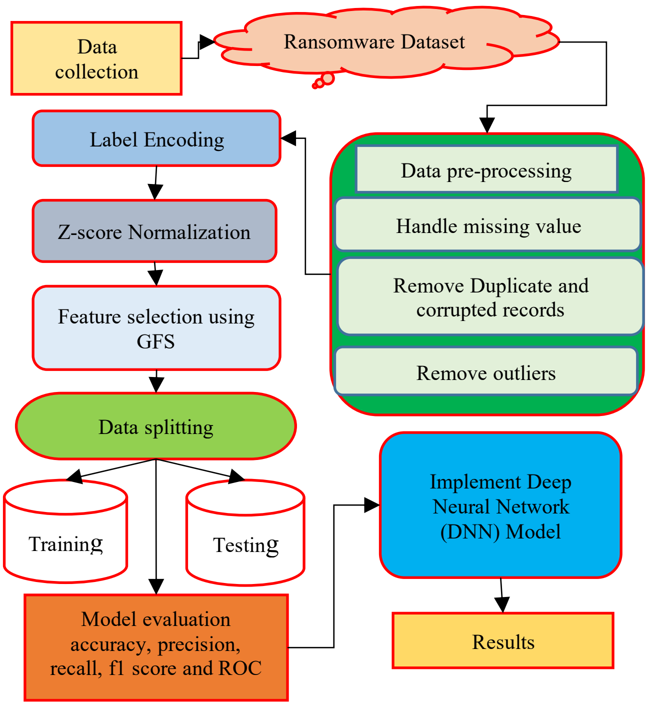

Fig. 1. Proposed flowchart for Ransomware Detection

A. *Data Gathering and Analysis*

A large and varied collection of 138,047 samples and 57 characteristics of ransomware was used. The selection of this dataset is done in such a way that it provides a fair portrayal of both the ransomware and the innocuous executables, and it is through this selection that the model can be able to achieve generalizability to different families of ransomware and to different types of innocent software. The following data visualizations, which include bar graphs and utilize heatmaps, feature correlations and attack dispersion were examined, etc:

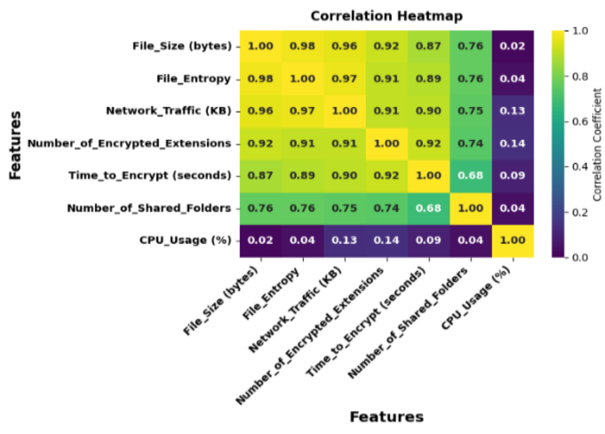

Fig. 2. Correlation Heatmap of Ransomware Dataset

Figure 2 shows that ransomware-related characteristics such network traffic, file size, and file entropy, and time of encryption have strong positive relationships, which implicates the high level of interdependence of these features during attack behavior. Conversely, there is very low correlation between CPU usage and other features, which implies that it provides independent data to the model of detection.

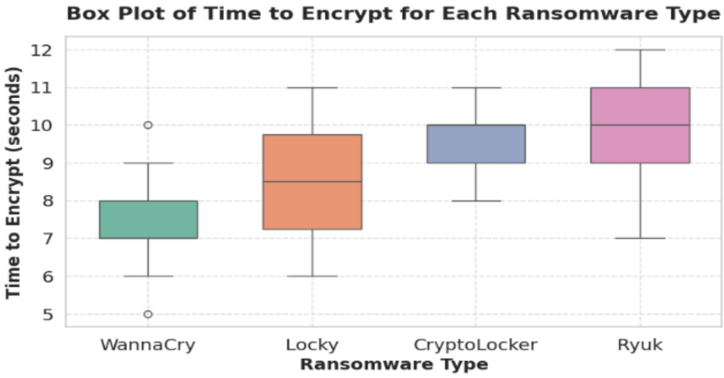

Fig. 3. Boxplot for each ransomware type

Figure 3 compares the time spent encrypting files with the ransomware of the different types, and it is possible to notice significant differences in the encryption behavior. Ryuk and Crypto Locker have a longer median encryption time, whereas WannaCry and Locky have a comparatively shorter encryption time with greater variability.

B. Data Pre-processing

Data preparation used Ransomware Dataset, with data concatenation, cleaning, and feature selection. The reprocessing phase involved handling missing values, duplicates, corrupted records, outliers, data labelling, and normalization. The reprocessing main steps are listed as follows:

- **Handle missing value:** Imputation mean substitution (Simple Imputer (strategy = mean)) was used to fill in missing values for normally distributed features, and Simple Imputer (strategy = median) was used for skewed features.
- **Remove Duplicate and corrupted records:** Python was used to thoroughly clean the dataset: faulty or unnecessary PE files were filtered using a validation flag (df[df['is_valid_pe'] = = True]), and duplicate entries were eliminated with df.drop_ duplicates().
- **Remove outliers:** Statistical methods were used to identify the outliers and those were eliminated to minimize noise in the data. The step is used to enhance the stability of the models and improve the overall prediction accuracy.
- **Label Encoding:** The labels of the categorical classes were transformed into numerical form to enable them to be used in the ML algorithms. This transformation takes care of the effective model training and correct classification performance.

C. Z-score Normalization

Data Normalization is the process of transforming or standardizing data to achieve a similar distribution. It has employed the z-score normalization approach, which has a mean of 0 and a standard deviation of 1. The values that are centered on the average value are converted using the unit standard deviation using this scaling approach. Equation (1) defines the z-score normalization.

$$E' = \frac{E-\overline{M}}{\sigma_M} \quad (1)$$

Where $\overline{M}$ is the mean, σ_M is the standard deviation, and E' and E are new and old for each data item.

D. Feature selection using GFS

The feature selection process, which seeks to identify the most important characteristics in a dataset, is one of the most important aspects of ML. GFS represents the graph-based feature selection model in the form of nodes within the graph, and the relationship or correlation between the nodes is represented as the edges. This method effectively models a complex interaction of features and finds the best subsets leading to better model performance particularly with high-dimensional data sets.

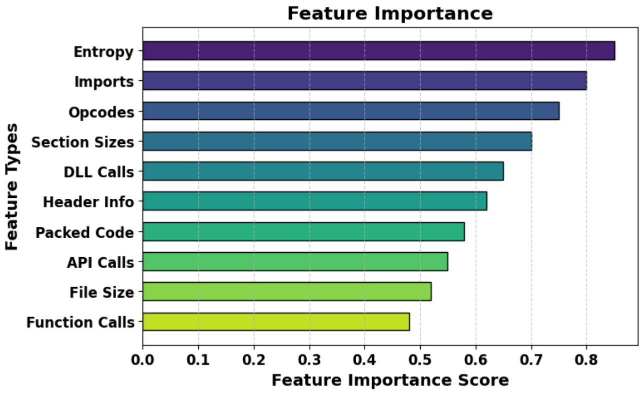

Fig. 4. Plot feature importance score

The feature importance scores as used in the model are shown in Figure 4, which shows the attributes that contribute most to the model predictions. The highest is Entropy, Imports and Opcodes respectively, which implies that they have important part of the decision-making process. The impact of other aspects like File Size and Function Calls is minimal. Visual interpretation. The horizontal bar chart helps to visually highlight the relative importance of every feature, which is useful to interpret and enhance the model transparency.

E. Data Splitting

The splits of the train and test sets were not random but stratified, with a 70:30 ratio, and the original class distribution was maintained in both sets to avoid biased, unreliable model testing.

F. Proposed Deep Neural Networks (DNNs) Model

A common DL method among academics is the DNN. The input, hidden, and output layers make up the DNN's network structure, and each layer is completely linked. Each neuron in the next layer is linked to every other neuron; however, these neurons are not connected to each other across layers[29]. An activation function that operates on the output following each network layer strengthens the impact of network learning. Consequently, DNN may also be viewed as a large perceptron composed of several perceptrons. For instance, the following formula may be used to calculate the ith layer forward propagation Equation (2):

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

where the input value is denoted by x, the weight coefficient matrices by w, and the bias vector by b. ReLU is typically employed as an activation function in a multi-class network; the formula is as follows Equation (3).

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

In order to optimize the network structure, the loss function computes the backpropagation of the network through the training samples' output loss and assesses the loss function. Often used as the loss function in classification issues, cross-entropy has the following Equation(4):

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

where N represents number of categories, M number of input data sets, yi the probability that the classification i will fall into the real category, and pi the probability of doing so. The DNN was set up with 4,000 epochs, batch size 32, Adam optimizer (learning rate 0.001), and ReLU activation. To handle binary classification, cross-entropy loss was used, and dropout (0.2) was used to avoid overfitting. Stability was increased via initialization, and optimal training was assured by early termination based on validation loss.

G. Evaluation Metrics

The performance measures that were used in assessing the effectiveness of there were several performance measures in the suggested model. A confusion matrix was used to summarize the classification by summarizing the correct and incorrect predictions of all the classes. This matrix was used to identify which metrics were significant: TP, FP, TN, and FN stand for true positives, false positives, and false negatives, respectively. To compute the standard assessment metrics, the following values were utilized: prec, rec, acc, and F1:

Accuracy: The percentage of instances from the input samples (dataset) that the trained model correctly predicted. It is shown as Equation(5)-

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

Precision: The accuracy of a model is calculated as the ratio of correctly predicted positive cases to all positive instances. Accuracy demonstrates the classifier's ability to predict the positive classifications is represented by Equation (6)-

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \quad (6)$$

ISSN:3049-0952
www.jsetms.com

Recall: This measure is the percentage of accurately predicted favorable outcomes for each case that should have been successful. In mathematics, it is represented as Equation (7)-

$$Recall = \frac{\mathrm{TP}}{TP+FN} \tag{7}$$

F1 score: It helps balance memory and accuracy by combining the harmonic mean of the two metrics. It has a range of [0, 1]. In terms of mathematics, it is Equation (8)-

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \tag{8}$$

Receiver Operating Characteristic Curve (ROC): A visual depiction that shows the percentage of cases correctly identified as positive across various decision cut-off points, compared with those wrongly classified as positive. TPR is also known as recall or sensitivity, whereas FPR is equivalent to 1-specificity.

IV. RESULTS AND DISCUSSION

The tests were carried out on the laptop, which had an Intel Core i9-14900HX processor, 32 GB RAM, and an NVIDIA RTX 4070 graphics card (8 GB VRAM) and ran in Python environment in a Jupyter notebook. After being trained on the ransomware dataset, the proposed DNN model was assessed using common performance measures, including acc, prec, rec, and F1, as indicated in Table I. The findings show the model's exceptional detection, showing that it correctly classifies almost all samples with an accuracy of 99.76%.

TABLE I. CLASSIFICATION RESULTS OF THE PROPOSED DNN MODEL FOR RANSOMWARE DETECTION

Matrix	Proposed DNN Model
Accuracy	99.76
Precision	99.76
Recall	99.76
F1-score	99.76

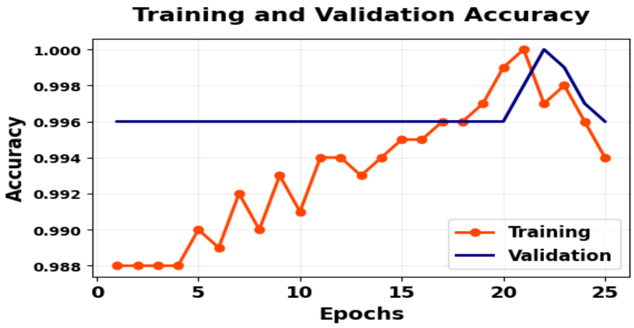

Fig. 5. Training and Validation Accuracy Curve for the DNN Model

The training and validation accuracy curves for DNN model over epochs are displayed in Figure 5. The accuracy of the training steadily increases to about 98.9% in the first epochs but to about 99.6-99.7% in the later epochs, meaning that the learning behaviour has stabilized. The accuracy of the validation is relatively stable, comparable to the training curve, at 99.6 to 100% , and indicates good generalization results with no apparent overfitting.

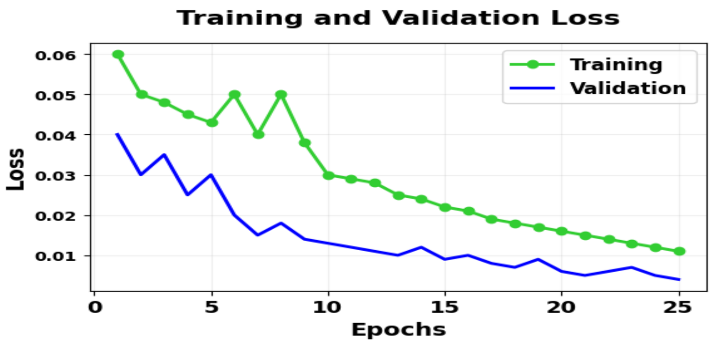

Fig. 6. Training and Validation Loss Curve for DNN model

The DNN model's loss curve in Figure 6 demonstrates efficient learning and convergence as the training and validation losses decrease over a period of 25 epochs. The training loss begins at a high point and reduces steadily, whereas the validation loss is lower at all times, indicating good generalization and low levels of overfitting. On the whole, the model has consistent performance and enhanced accuracy with an increase in training.

 ISSN:3049-0952
www.jsetms.com

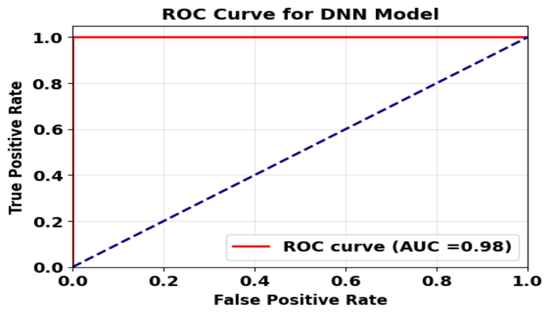

Fig. 7. ROC Curve for DNN model

The ROC curve of DNN model, as depicted in Figure 7, shows that it has a strong ability to classify. The curve is steeply upward, ascending to the top-left, and is a sensitive curve with low false positives. The AUC value of 0.98 indicates nearly flawless discrimination between positive and negative classifications. The diagonal line represents random performance, to which the DNN is plainly better.

A. Comparative analysis

The suggested DNN model was compared to current ML and DL models in order to assess its efficacy, as shown in Table II. The accuracy of KNN model is 83.9%, whereas the VGG-16 model performs at 90.5%, XGBoost also improves the results, achieving 94.1% accuracy, whereas the Logistic Regression achieves a high recall of 96%. However, the proposed DNN has the best performance, which proves its better performance in ransomware detection than all other tested models.

TABLE II. COMPARISON OF DIFFERENT ML MODELS FOR RANSOMWARE DETECTION

Model	Accuracy	Precision	Recall	F1-score
KNN[30]	83.9	83.8	83.9	83.8
VGG 16[31]	90.5	89.73	87.43	88.74
XGBoost[32]	94.1	92.5	90.8	91.6
LR[33]	96	89	96	89
Proposed DNN	99.76	99.76	99.76	99.76

The Deep Neural Network (DNN) model proposed has a number of strengths in ransomware detection. The dataset's intricate, non-linear relationships are well represented by its design, which makes it possible to understand the data more effectively than traditional machine learning models. By removing redundancy and ensuring that the features stay relevant, Graph-Based Feature Selection (GFS) is incorporated, improving detection accuracy. Z-score normalization also provides consistent data scaling, which results in faster convergence and improved generalization. The tight alignment of the model exhibits a good level of stability with little overfitting, according to training and validation loss and accuracy curves. Altogether, DNN model offers higher precision, strength and reliability and reaches almost perfect performance, in terms of ransomware detection.

The proposed DNN model has limitations even though it is outstanding in its performance. Also, the method has a high computational cost to train and therefore might not be scalable to resource-constrained settings. The future research can be dedicated to further optimization of the model to work with lightweight architectures to empower real-time detection. Improving the interpretability of model-generated predictions by explainable AI techniques to aid in better decision-making in cybersecurity applications.

V. CONCLUSION AND FUTURE STUDY

Ransomware detection systems can identify the threat faster and give the victim time to act before it is too late. Ransomware discovery will help prevent the loss of important data. Some users do not access their original data once again after an intrusion. The suggested DNN model performed remarkably in detecting ransomware, whereby accuracy, precision, recall, and F1-score were 99.76 respectively. GFS integration provided better relevance of features, whereas z-score normalization provided optimal data scaling and convergence. In addition to this, the ROC curve had an AUC of 0.98 showing near perfection of discrimination. The proposed DNN was confirmed to have superior performance by comparing it with other models, which include KNN, VGG-16, XGBoost, and Logistic Regression. On the whole, the research proves that the DNN framework is an effective, stable, and scalable solution in the detection of ransomware threats within any contemporary cybersecurity framework.

REFERENCES

[1] R. Patel, "Automated Threat Detection and Risk Mitigation for ICS (Industrial Control Systems) Employing Deep Learning in Cybersecurity Defense," *Int. J. Curr. Eng. Technol.*, vol. 13, no. 06, pp. 584–591, 2023, doi: 10.14741/ijcet/v.13.6.11.

[2] V. M. L. G. Nerella, K. K. Sharma, S. Mahavratayajula, and H. Janardhanan, "A Machine Learning Framework for Cyber Risk Assessment in Cloud-Hosted Critical Data Infrastructure," *J. Inf. Syst. Eng. Manag.*, vol. 10, no. 4, pp. 2409–2421, 2025, doi: 10.52783/jisem.v10i4.12804.

[3] S. Singh, "Advancing Network Security in 5G: Leveraging the 5G-NIDD Dataset for Intrusion Detection and Mitigation," in *2025 IEEE 12th International Conference on Cyber Security and Cloud Computing (CSCloud)*, IEEE, Nov. 2025, pp. 1–6. doi: 10.1109/CSCloud66326.2025.00055.

[4] V. K. Bollu, "Threat Landscape in Artificial Intelligence Systems: Taxonomy, Attack Vectors and Security Implications," *World J. Adv. Res. Rev.*, vol. 29, no. 1, pp. 285–294, 2026, doi: 10.30574/wjarr. 2026.29.1.0007.

[5] S. A. Pushkala, "Financial Fraud Identification Using Graph Neural Network And LSTM With Autoencoder-Based Data Refinement," *J. Int. Cris. Risk Commun. Res.*, vol. 9, no. 1, 2026, doi: 10.63278/jicrcr.vi.3615.

[6] S. Chatterjee, "Integrating Identity and Access Management for Critical Infrastructure: Ensuring Compliance and Security in Utility Systems," *Int. J. Innov. Res. Creat. Technol.*, vol. 8, no. 2, pp. 1–8, 2022.

[7] N. K. Prajapati, "Federated Learning for Privacy-Preserving Cybersecurity: A Review on Secure Threat Detection," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 5, no. 4, pp. 520–528, Apr. 2025, doi: 10.48175/IJARSCT-25168.

[8] V. Verma, "Security Compliance and Risk Management in AI-Driven Financial Transactions," *Int. J. Eng. Sci. Math.*, vol. 12, no. 7, July, pp. 107–121, 2023.

[9] T. A. Khan *et al.*, "Multi-Source Cyber Intrusion Detection Using Ensemble Machine Learning," *J. Comput. Sci.*, vol. 21, no. 1, pp. 111–123, Dec. 2024, doi: 10.3844/jcssp. 2025.111.123.

[10] A. R. Bilipelli, "Forecasting the Evolution of Cyber Attacks in FinTech Using Transformer-Based Time Series Models," *Int. J. Res. Anal. Rev.*, vol. 10, no. 3, pp. 383–389, 2023.

[11] J. B. Mehta, "Autonomous Patch Validation For Zero-Day Exploits In Enterprise Clouds," *Int. J. Appl. Math.*, vol. 38, no. 4s, pp. 1270–1285, August, 2025, doi: 10.12732/ijam.v38i4s.685.

[12] R. Lingam, "Zero-Trust Architectures for Secure DevOps Automation in Enterprise AI Systems," *Milestone Trans. Artif. Intell.*, vol. 1, no. 1, pp. 18–33, Jan, 2026.

[13] D. W. Fernando, N. Komninos, and T. Chen, "A Study on the Evolution of Ransomware Detection Using Machine Learning and Deep Learning Techniques," *IoT*, vol. 1, no. 2, pp. 551–604, Dec. 2020, doi: 10.3390/iot1020030.

[14] V. Prajapati, "Role of Identity and Access Management in Zero Trust Architecture for Cloud Security : Challenges and Solutions," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 5, no. 3, pp. 6–18, 2025, doi: 10.48175/IJARSCT-23902.

[15] V. Shewale, "Demystifying the MITRE ATT&CK Framework: A Practical Guide to Threat Modeling," *J. Comput. Sci. Technol. Stud.*, vol. 7, no. 3, pp. 182–186, May 2025, doi: 10.32996/jcsts.2025.7.3.20.

[16] G. Sarraf, "AI-Enhanced Critical Infrastructure Defense: Protecting SCADA and ICS Networks Through Intelligent Monitoring," *Int. J. Curr. Eng. Technol.*, vol. 14, no. 6, pp. 533–540, 2024, doi: 10.14741/ijcet/v.14.6.16.

[17] S. Pawar, G. Patil, K. Patel, P. Pawar, S. Khedkar, and B. More, "Falsified News Detection Using Deep Learning Approach," in *2021 Asian Conference on Innovation in Technology (ASIANCON)*, 2021, pp. 1–5. doi: 10.1109/ASIANCON51346.2021.9544585.

[18] S. Narang and A. Gogineni, "Zero-Trust Security in Intrusion Detection Networks: An AI-Powered Threat Detection in Cloud Environment," *Int. J. Sci. Res. Mod. Technol.*, vol. 4, no. 5, pp. 60–70, jun. 2025, doi: 10.38124/ijsrmt.v4i5.542.

[19] V. Shah, "Traffic Intelligence In Iot And Cloud Networks: Tools For Monitoring, Security, And Optimization," *Int. J. Recent Technol. Sci. Manag.*, vol. 5, 2024, doi: 10.10206/IJRTSM.2025894735.

[20] R. Rao Thallada and N. Alapati, "Privacy and Cybersecurity Convergence: GRC Controls for Data Protection," Journal *Bus. Manag. Stud.*, vol. 8, no. 5, pp. 42–48, March, 2026, doi: 10.32996/jbms.

[21] G. Sarraf, "Autonomous Ransomware Forensics: Advanced ML Techniques for Attack Attribution and Recovery," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 3, no. 3, pp. 1377–1390, Jul. 2023, doi: 10.48175/IJARSCT-11978W.

[22] C. H. M. Souza and D. M. Batista, "On the Use of Machine Learning for Modern IoT ELF Malware Detection," in *2025 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, 2025, pp. 1–6. doi: 10.1109/LA-CCI66231.2025.11270436.

[23] R. Kipanga and F. Khennou, "Leveraging Feature Selection and Deep Learning for Accurate Malware and Ransomware Detection in PE Files," in *ISDFS 2025 - 13th International Symposium on Digital Forensics and Security*, 2025. doi: 10.1109/ISDFS65363.2025.11012017.

[24] A. Polamarasetti, "Research developments, trends and challenges on the rise of machine learning for detection and classification of malware," in *Intelligent Computing and Emerging Communication Technologies, ICEC 2024*, 2024. doi: 10.1109/ICEC59683.2024.10837413.

[25] R. P. Baksi, V. Nalka, and S. Upadhyaya, "Apt Detection of Ransomware - An Approach to Detect Advanced Persistent Threats Using System Call Information," in *Proceedings - 2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom/BigDataSE/CSE/EUC/iSCI 2023*, 2023. doi: 10.1109/TrustCom60117.2023.00221.

[26] N. Aljubory and B. M. Khammas, "Hybrid Evolutionary Approach in Feature Vector for Ransomware Detection," in *International Conference on Intelligent Technology, System and Service for Internet of Everything, ITSS-IoE 2021*, 2021. doi: 10.1109/ITSS-IoE53029.2021.9615344.

[27] M. Basnet, S. Poudyal, M. H. Ali, and D. Dasgupta, "Ransomware detection using deep learning in the SCADA system of an electric vehicle charging station," in *2021 IEEE PES Innovative Smart Grid Technologies Conference - Latin America, ISGT Latin America 2021*, 2021. doi: 10.1109/ISGTLatinAmerica52371.2021.9543031.

[28] S. Singamsetty, "Accelerating data engineering efficiency with self-learning AI algorithms," *Int. J. Comput. Artif. Intell.*, vol. 6, no. 1, pp. 195–199, Jan. 2025, doi: 10.33545/27076571.2025.v6.i1c.154.

[29] B. Jeganathan, "High-Accuracy Mixed-Type Wafer Defect Classification Using a Custom Alex Net Architecture on the Mixed WM38 Dataset," *Int. J. Res. Innov. Appl. Sci.*, vol. 10, no. 6, July, pp. 735–748, 2025, doi: 10.51584/IJRIAS.2025.10060057.

[30] M. M. Singh, K. Selvaraj, and Z. Wei, "Enhanced detection of Android ransomware families using machine learning and network traffic analysis," *Bull. Electr. Eng. Informatics*, vol. 14, no. 4, pp. 2987–2996, Aug. 2025, doi: 10.11591/eei.v14i4.9485.

[31] A. Singh, Z. Mushtaq, H. A. Abosaq, S. N. F. Mursal, M. Irfan, and G. Nowakowski, "Enhancing Ransomware Attack Detection Using Transfer Learning and Deep Learning Ensemble Models on Cloud-Encrypted Data," *Electronics*, vol. 12, no. 18, p. 3899, Sep. 2023, doi: 10.3390/electronics12183899.

[32] S. Satpathy and P. K. Swain, "Graph-contrast ransomware detection (GCRD) with advanced feature selection and deep learning," *Discov. Comput.*, 2025, doi: 10.1007/s10791-025-09651-w.

[33] A. P. Ferreira, C. Gupta, P. R. M. Inácio, and M. M. Freire, "Behavior-based Malware Detection in Mobile Android Platforms Using Machine Learning Algorithms," *J. Wirel. Mob. Networks, Ubiquitous Comput. Dependable Appl.*, vol. 12, no. 4, pp. 62–88, 2021, doi: 10.22667/JOWUA.2021.12.31.062.