**TELKOMNIKA Telecommunication Computing Electronics and Control**
Vol. 24, No. 3, June 2026, pp. 933~944
ISSN: 1693-6930, DOI: 10.12928/TELKOMNIKA.v24i3.27833 

# Machine learning and deep learning for ransomware detection via feature decontamination

**Sriyanto[1], Chairani Fauzi[1], Mohd Faizal Abdollah[2], Zuriati[3]**
[1]Department of Informatics Engineering, Institute of Informatics and Business Darmajaya, Bandar Lampung, Indonesia
[2]Department of Cybersecurity, Faculty of Artificial Intelligence and Cyber Security, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia
[3]Department of Internet Engineering Technology, Politeknik Negeri Lampung, Bandar Lampung, Indonesia

**Article Info**

***Article history:***

Received Jan 27, 2026
Revised Mar 16, 2026
Accepted Mar 29, 2026

***Keywords:***

Cybersecurity
Feature decontamination
Intrusion detection
Network traffic analysis
Ransomware
Synthetic minority over-sampling technique

**ABSTRACT**

The continuous escalation of ransomware attacks poses a severe risk to network infrastructure and data integrity, highlighting the urgent requirement for dependable detection systems. This paper presents a comparative analysis of deep learning (DL) and machine learning (ML) techniques for identifying ransomware traffic using the UNSW-NB15 dataset. A significant obstacle in many intrusion detection investigations is feature contamination, where specific attributes inadvertently leak label data or reflect post-incident statistics, resulting in inflated and overly optimistic performance evaluations. To mitigate this concern, a feature decontamination protocol is implemented to isolate 29 reliable attributes, followed by the application of the synthetic minority over-sampling technique (SMOTE) to address the issue of class imbalance. Empirical results demonstrate that the random forest (RF) model achieves superior performance, reaching an accuracy of 0.9027 and a recall of 0.9507. Among the DL candidates, the multi-layer perceptron (MLP) delivers the most competitive outcomes with an accuracy of 0.8859 and an F1-score of 0.8996. These results suggest that ensemble-based ML frameworks offer more effective and computationally efficient ransomware detection when applied to decontaminated tabular datasets.



***Corresponding Author:***

Sriyanto
Department of Informatics Engineering, Institute of Informatics and Business Darmajaya
ZA. Pagar Alam St. No.93, Gedong Meneng, Rajabasa, Bandar Lampung, Lampung, Indonesia
Email: sriyanto@darmajaya.ac.id

## 1. INTRODUCTION

The escalating intensity and sophistication of cyberattacks have established ransomware as a major threat to the stability of contemporary network infrastructures [1], [2]. As a form of crypto virology, ransomware paralyzes systems by encrypting data for ransom, executing operational sabotage, or causing data destruction [3]. This threat leverages complex network traffic dynamics within seconds, necessitating the integration of artificial intelligence (AI) and edge computing to provide accelerated detection at the data source [4]. The adaptive nature of these threats requires the development of intrusion detection systems (IDS) that offer high accuracy and operational reliability across large-scale, heterogeneous network environments [5], [6]. In modern telecommunications and distributed computing frameworks, ransomware detection has emerged as a fundamental mechanism for ensuring service availability and secure data transmission.

Machine learning (ML) and deep learning (DL) methodologies have become the gold standard for intrusion detection. ML models, including decision trees (DT), random forests (RF), and extreme gradient

boosting (XGBoost), are frequently utilized due to their efficiency in processing tabular data. Conversely, DL architectures like multi-layer perceptron (MLP), 1D-convolutional neural networks (1D-CNN), and long short-term memory (LSTM) are adept at identifying complex non-linear patterns and executing automated feature extraction [7]. Comparative studies by [8]-[10] emphasize the persistent debate regarding the efficacy of these two paradigms. While DL models offer robust representation learning capabilities, their deployment is often limited by substantial computational requirements, particularly in internet of things (IoT) ecosystems [11]. Some contemporary studies have investigated graph-based modeling to interpret topological relationships in network traffic [12]. Nevertheless, for structured tabular datasets, tree-based models often surpass DL approaches in both computational efficiency and predictive accuracy [13], [14]. These observations suggest that the effectiveness of ML or DL models is heavily contingent upon feature representation and dataset characteristics [8].

However, literature reviews reveal significant methodological challenges when utilizing publicly available cybersecurity datasets. The UNSW-NB15 dataset, designed to represent modern network traffic patterns, has been reported to contain issues such as data imbalance, class overlaps [6], and feature redundancy that may hinder reliable classification [15], [16]. More importantly, several studies report that some attributes may implicitly encode label-related information, creating dataset artifacts that bias model evaluation. These issues are further exacerbated when models are trained on features that contain post-event statistics or indirect label indicators, a phenomenon commonly referred to as feature contamination. The presence of such contaminated features can create trivial separability and produce overly optimistic evaluation results, which do not reflect the true capability of the detection system in real-world environments [6], [8]. Furthermore, the absence of standardized feature selection protocols across intrusion detection studies makes model generalization across different datasets and network environments increasingly difficult [17], leading to the continuous development of new datasets derived from real network traffic [18].

Despite the widespread application of the UNSW-NB15 dataset in the field of intrusion detection, the majority of current research tends to prioritize the enhancement of classification metrics or the introduction of novel learning frameworks, often overlooking the critical issues of potential data leakage and feature contamination. Consequently, the claimed performance advantages of specific models, notably DL methods might stem from concealed dataset artifacts instead of authentic representational strengths. This constraint underscores a significant methodological void in contemporary intrusion detection studies, where the evaluation of model performance occurs without verifying the integrity of the feature space employed during training and testing [19].

The primary contributions of this work are outlined as follows. First, a feature decontamination protocol is executed to remove any attributes that might inadvertently leak label-related data within the UNSW-NB15 dataset. Second, the synthetic minority over-sampling technique (SMOTE) is utilized to mitigate class imbalance and enhance the sensitivity of ransomware detection. Third, a rigorous comparative analysis is performed between tree-based ML algorithms (DT, RF, and XGBoost) and DL models (MLP, 1D-CNN, and LSTM) using the cleaned feature set.

The importance of this research is rooted in established a more pragmatic evaluation framework for ransomware identification within computing and telecommunication infrastructures. The results deliver actionable insights for choosing efficient detection models that maintain an equilibrium between computational overhead and predictive accuracy, a factor vital for real-time security surveillance in contemporary network settings [7], [11].

## 2. METHOD

### 2.1. Research framework

This study is designed as a quantitative comparative study to evaluate the performance of ML and DL approaches in ransomware traffic classification [1]. The experimental framework aims to provide a systematic evaluation of different learning paradigms under identical preprocessing conditions to ensure a fair comparison of model performance. The framework is systematically structured to address model validity challenges on public datasets that often contain inherent biases [6]. The primary focus is to ensure a fair and realistic evaluation by controlling data leakage and feature contamination [8].

To improve reproducibility and experimental transparency, the research follows a structured pipeline consisting of dataset acquisition, preprocessing, feature decontamination, class balancing, model implementation, and performance evaluation. The research stages follow the workflow illustrated in Figure 1, covering dataset acquisition, preprocessing (feature decontamination and SMOTE) [7], comparative modeling, and evaluation using industry standard metrics [9], [10], [20]. This structured workflow ensures that each experimental stage is conducted sequentially to minimize bias and maintain consistency across all evaluated models.

All ML and DL models were evaluated under identical preprocessing conditions to ensure that performance differences arise from the learning algorithms rather than preprocessing bias or dataset artifacts. By applying the same preprocessing pipeline to all models, the study aims to provide a reliable benchmark for evaluating ransomware detection approaches using decontaminated network traffic features.

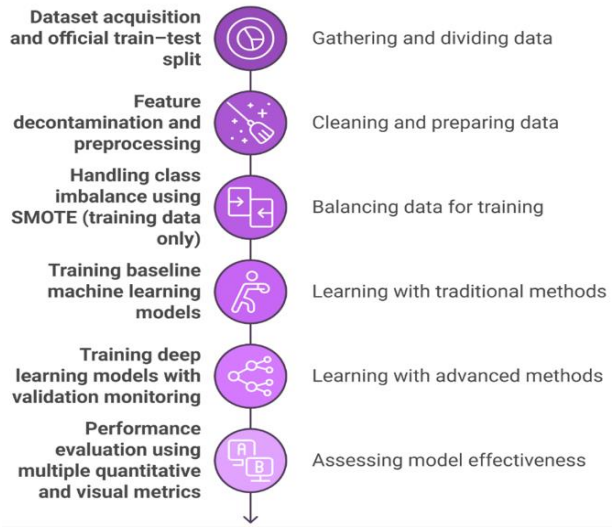



Figure 1. Research stages

### 2.2. Data acquisition

This research employs the UNSW-NB15 dataset, a contemporary cybersecurity benchmark established by the Australian Centre for Cyber Security at the University of New South Wales. While previous literature has noted certain constraints, UNSW-NB15 is still extensively utilized in intrusion detection studies due to its inclusion of authentic network traffic characteristics and varied attack scenarios [6], [15]. Researchers continue to favor this dataset because it offers a realistic depiction of modern network behavior and diverse threat profiles. To optimize computational performance during the training phase, the dataset is handled in Apache Parquet format. Although the raw UNSW-NB15 dataset comprises 49 features, many are unsuitable for model training as they function as identifiers or post-incident statistics that could skew evaluation results. Consequently, 36 technical attributes relevant to flow-based analysis were initially selected, omitting non-essential identifiers like timestamps and IP addresses.

These 36 attributes are organized into the following five categories:

a. Basic features: core metrics such as connection duration (dur), protocol (proto), state (state), and application service (service).
b. Content features: packet-level data including total bytes (sbytes and dbytes), packet counts (spkts and dpkts), time-to-live (sttl and dttl), and transmission rate (rate).
c. Time features: metrics involving jitter, inter-arrival time, packet loss (sloss and dloss), and load statistics (sload and dload).
d. Transmission control protocol (TCP) features parameters including round-trip time (RTT), window size, ackdat, synack, and base sequence numbers.
e. Statistical and last time management (LTM) features: attributes that capture connection frequency patterns to the same IP or service.

The dataset encompasses a total of 257,673 samples, divided into 175,341 training records and 82,332 testing records. Initial exploratory data analysis indicated a substantial disproportion between attack and normal traffic, which prompted the application of class-balancing and feature decontamination methods. To ensure experimental reproducibility and maintain alignment with existing research, this study adheres to the original training and testing split provided by the UNSW-NB15 developers.

### 2.3. Data preprocessing and feature decontamination

The preprocessing stage focuses on ensuring data integrity by applying a feature decontamination protocol (illustrated in Figure 2) to avoid trivial separability and hidden data leakage [6]. Feature decontamination aims to remove attributes that may implicitly contain label-related information or post-event

statistical summaries that could bias model evaluation. To identify potentially contaminated attributes, an exploratory analysis was conducted to examine the statistical relationships between features and class labels. Features that directly encode post-event traffic statistics or exhibit strong association with attack labels were considered potential sources of data leakage. These attributes may artificially simplify the classification task, leading to overly optimistic model performance.

The feature selection mechanism was conducted:

a. Leakage assessment, evaluating real-time feature availability to ensure that the model only uses attributes observable during actual network traffic monitoring.
b. Correlation-based leakage quantification, where highly correlated features with potential label leakage were examined using statistical correlation analysis.
c. Feature removal, eliminating LTM features and redundant statistical attributes identified as laboratory artifacts.

Through this process, features that represent aggregated connection statistics or indirectly encode attack outcomes were removed from the dataset. This step ensures that the learning models rely on genuine behavioral characteristics of network traffic rather than exploiting hidden dataset artifacts. Through this procedure, the feature set was reduced from 36 attributes to 29 clean features. This reduction is not merely dimensionality reduction but a deliberate strategy to eliminate contaminated features that could artificially inflate model performance.

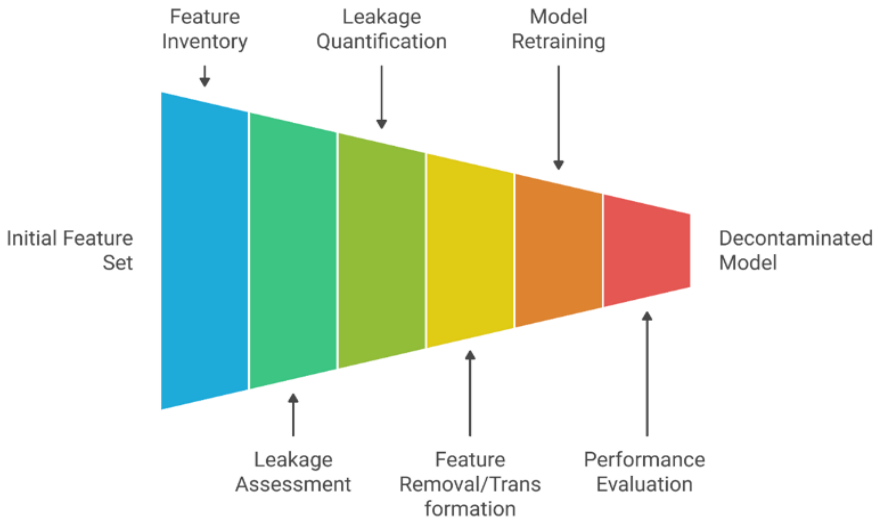



Figure 2. Feature decontamination process

The removed features and their justifications are detailed in Table 1.

Table 1. Feature decontamination results

| Category | Removed features | Retained features | Justification for elimination |
|---|---|---|---|
| Flow statistics | ct_state_ttl, ct_srv_src, ct_srv_dst | rate, sttl, dttl, sload, dload, sloss, dloss | Post-event features that trigger evaluation bias (data leakage) |
| TCP and protocol characteristics | is_ftp_login, is_sm_ips_ports | swin, dwin, stcpb, dtcpb, tcprtt, synack, ackdat | Simple indicator features that are trivial in nature |
| LTM and dynamics connections | ct_src_dport_ltm, ct_dst_sport_ltm | dur, spkts, dpkts, sbytes, dbytes, smean, dmean, sinpkt, dinpkt, sjit, djit, proto, service, state, trans_depth | Information redundancy and laboratory statistical artifacts |

Following decontamination, preprocessing continued with data standardization:

a. Label encoding, converting categorical features into numerical representations [20].
b. Min-max scaling, normalizing numerical features into the [0, 1] range to ensure stability during the weight update process in DL models [7].

### 2.4. Class balancing using SMOTE

The initial training dataset exhibited a strong dominance of attack classes over the normal class. Models trained on imbalanced data tend to be biased toward the majority class, increasing the risk of false negatives (FN), where critical ransomware attacks remain undetected [6], [21]. To address this issue, the SMOTE was applied to the training data. SMOTE generates synthetic samples by interpolating new instances between neighboring samples in the feature space, thereby improving the representation of minority classes without simple duplication. In this study, SMOTE was applied exclusively to the training dataset to avoid data leakage into the testing set. This ensures that model evaluation remains unbiased and reflects real-world performance.

In this study, SMOTE was implemented with $k = 5$, a commonly used value for generating synthetic minority samples in imbalanced classification problems. Previous studies have shown that combining SMOTE with ensemble-based classifiers, such as RF, can significantly improve detection sensitivity, particularly for minority attack classes [22]. The resulting balanced class distribution is visualized in Figure 3.

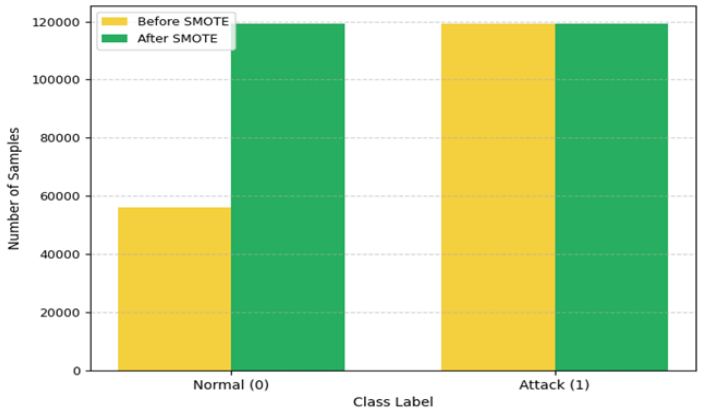



Figure 3. Class distribution before and after SMOTE

### 2.5. ML and DL implementation

The modeling phase compares ML algorithms and DL architectures to evaluate the effectiveness of the 29 decontaminated features in detecting ransomware activity [1]. All experiments were conducted using Python-based ML frameworks, primarily Scikit-learn for classical ML models and TensorFlow/Keras for DL architectures to ensure reproducible implementation of the models.

#### 2.5.1. ML architecture

Three tree based models were implemented due to their robustness in handling tabular data [13]:

a. DT, used as a baseline model to capture rule-based classification behavior.
b. RF, which aggregates multiple DT using bootstrap sampling to improve prediction stability and reduce variance [6].
c. XGBoost, a gradient boosting algorithm designed to optimize classification performance through iterative error correction [14].

#### 2.5.2. DL architecture

Three DL architectures were implemented to evaluate automatic feature extraction capability:

a. MLP, consisting of fully connected layers with rectified linear unit (ReLU) activation functions in hidden layers and Sigmoid activation in the output layer.
b. 1D-CNN, designed to capture local feature interactions through convolutional filters and pooling layers.
c. LSTM is implemented to evaluate potential sequential dependencies within the feature representation space. The LSTM model was included to investigate whether sequential learning mechanisms could capture hidden dependencies within aggregated network traffic features.

All DL models were trained using binary cross-entropy loss and the Adam optimizer with a learning rate of 0.001. Training was conducted for 50 epochs with a batch size of 32, and dropout regularization was applied to mitigate overfitting.

### 2.6. Evaluation metrics

Performance evaluation was conducted using standard metrics derived from the confusion matrix [9], [10]:

a. Accuracy, measuring the ratio of correct predictions over total samples.
b. Precision, measuring the correctness of positive predictions.
c. Recall (sensitivity) is considered the primary metric in ransomware detection because failing to detect an attack (FN) can result in severe system damage.
d. F1-score, the harmonic mean of precision and recall.
e. Area under the curve-receiver operating characteristic (AUC-ROC), measures the model's ability to discriminate between classes across multiple threshold settings.

These evaluation metrics provide a comprehensive assessment of model performance beyond simple accuracy, particularly when dealing with imbalanced cybersecurity datasets.

## 3. RESULTS AND DISCUSSION

### 3.1. Performance evaluation of ML models

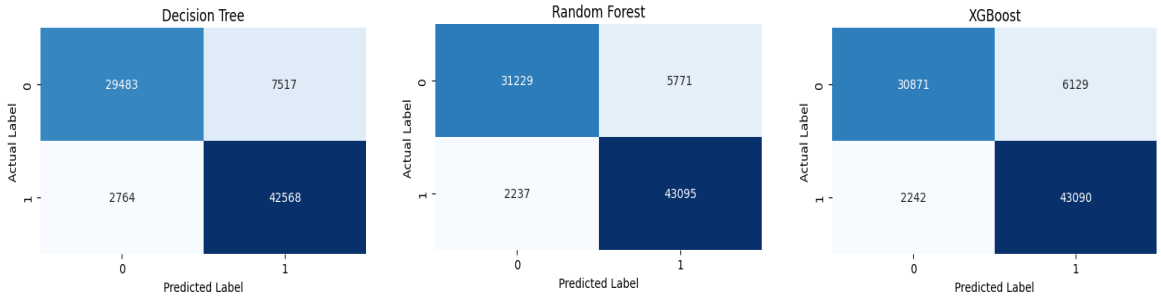
This section presents a comprehensive evaluation of the baseline ML models in detecting ransomware-related network traffic using the UNSW-NB15 dataset. The evaluation focuses on model behavior after applying feature decontamination and class balancing, ensuring that the observed performance reflects the models' ability to learn meaningful traffic patterns rather than exploiting dataset artifacts. The evaluated ML algorithms include DT, RF, and XGBoost.

#### 3.1.1. Confusion matrix analysis and ML evaluation metrics

The classification capabilities of the baseline ML models are visually represented through the confusion matrices shown in Figure 4, consisting of three subfigures: (a) DT, (b) RF, and (c) XGBoost, allowing direct comparison of misclassification patterns across models.

As illustrated in Figure 4(a), the DT model demonstrates reasonable detection capability but produces a relatively higher number of false positives (FP). This behavior is reflected in Table 2, where DT records the lowest precision value (0.8499). The result indicates that single-tree architectures tend to overfit heterogeneous network traffic distributions, particularly when the feature space contains complex interactions among traffic attributes.

In contrast, Figures 4(b) and (c) show that ensemble-based models establish more stable decision boundaries. According to Table 2, RF achieves the highest true positive (TP) rate, with a recall of 0.9507. This high recall capability is particularly critical for ransomware detection, as missing even a single attack instance can result in large-scale data encryption and system disruption.




(a) (b) (c)

Figure 4. Confusion matrix of ML models: (a) DT, (b) RF, and (c) XGBoost

Meanwhile, XGBoost produces a strong F1-score of 0.9115, indicating balanced performance between precision and recall. The superior performance of ensemble-based models is consistent with previous studies showing that bagging and boosting techniques are highly effective in handling structured tabular datasets with heterogeneous statistical patterns [7]. Further inspection of the confusion matrices also reveals that most misclassification cases occur between benign and attack traffic instances that share similar statistical characteristics, such as packet rate and connection duration. This overlap indicates that certain network traffic patterns exhibit behavioral similarities, which may result in occasional false-positive

predictions. Nevertheless, the overall number of FN remains relatively low, particularly in the RF model, demonstrating strong capability in identifying ransomware-related traffic patterns. The results confirm that ensemble-based ML models significantly outperform single-tree classifiers. After feature decontamination, the models are forced to rely on genuine behavioral patterns of network traffic rather than trivial dataset artifacts, producing a more realistic evaluation of ransomware detection capability.

Table 2. ML model performance

| Model | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| DT | 0.8751 | 0.8499 | 0.9390 | 0.8923 |
| RF | 0.9027 | 0.8819 | 0.9507 | 0.9150 |
| XGBoost | 0.8983 | 0.8755 | 0.9505 | 0.9115 |

### 3.1.2. Discriminative characteristics through ROC curve analysis

The discriminative ability of the ML models is further evaluated using the ROC curve in Figure 5. The ROC curves show that RF and XGBoost consistently achieve higher TP rates across multiple classification thresholds than the DT model. The RF curve is positioned closest to the top-left corner, indicating superior discriminative capability. In IDS, a higher AUC value is particularly desirable because it indicates the model's ability to maintain high detection rates while minimizing false alarms. Reducing false alarms is essential in operational network environments to prevent alarm fatigue among security analysts [9], [10]. The strong ROC performance of ensemble models also suggests that combining multiple DT improves the robustness of classification boundaries when dealing with heterogeneous network traffic features. This behavior is particularly beneficial for IDS deployed in operational telecommunication networks, where maintaining high detection sensitivity while controlling false alarms is essential for reliable security monitoring.

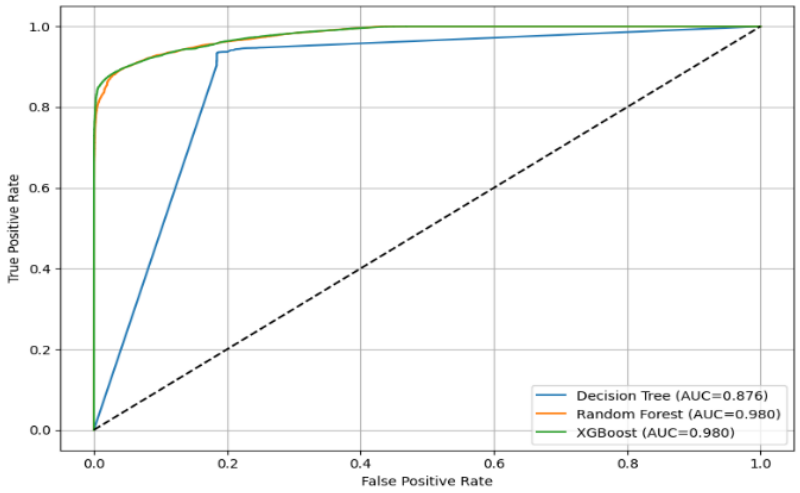


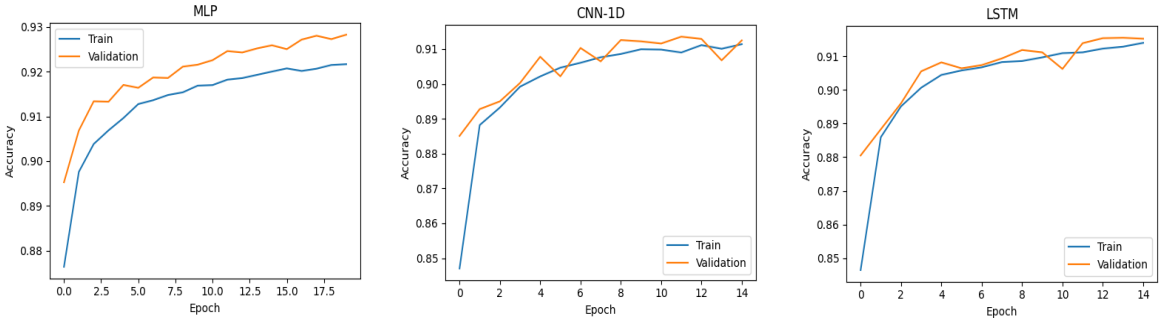

Figure 5. ROC curve

## 3.2. Performance evaluation of DL models

This section evaluates the performance of DL architectures, including MLP, 1D-CNN, and LSTM, in detecting ransomware traffic using the decontaminated feature set.

### 3.2.1. DL training dynamics and convergence analysis

The training stability of DL models is analyzed using accuracy curves shown in Figure 6, which consist of three subfigures representing: (a) MLP, (b) 1D-CNN, and (c) LSTM models. As illustrated in Figure 6(a), the MLP model demonstrates stable convergence between training and validation accuracy, indicating consistent learning behavior without significant overfitting. Figure 6(b) shows the training dynamics of the 1D-CNN model, with the validation accuracy fluctuating slightly before stabilizing as training progresses. Similarly, Figure 6(c) presents the convergence pattern of the LSTM model, which also achieves stable learning behavior after several training epochs.



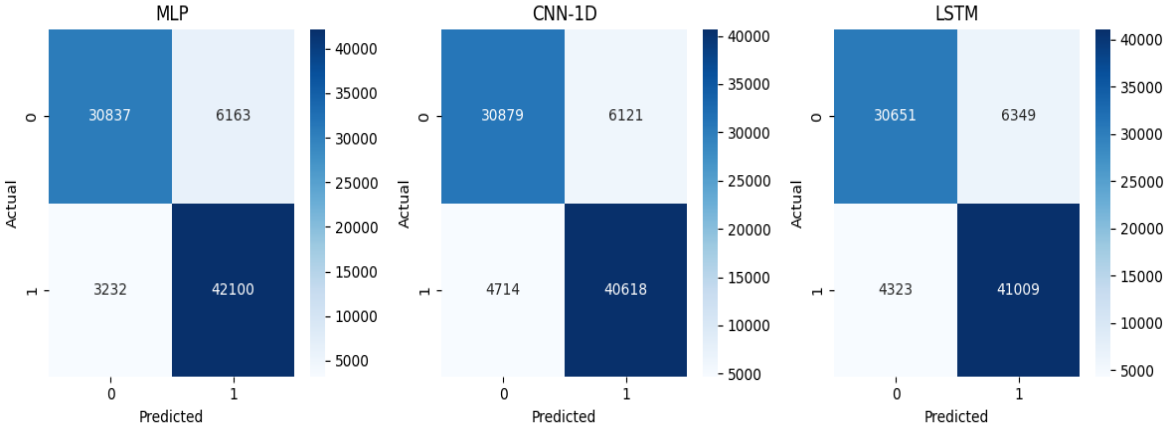

(a) (b) (c)

Figure 6. Training vs validation accuracy curves of DL models: (a) MLP, (b) 1D-CNN, and (c) LSTM

Overall, the models demonstrate stable convergence behavior during training. The use of min-max normalization and the Adam optimizer helps stabilize gradient updates and prevents numerical instability during optimization. However, the validation curves indicate that deeper architectures such as 1D-CNN and LSTM do not produce significant performance improvements compared to the simpler MLP model. This suggests that the tabular structure of the dataset does not strongly benefit from architectures designed for spatial or sequential feature learning. As a result, simpler neural architectures such as MLP can provide more stable learning behavior without requiring extensive computational resources.

### 3.2.2. Confusion matrix analysis and DL evaluation metrics

The confusion matrices presented in Figure 7 illustrate the distribution of classification errors across DL models, including (a) MLP, (b) 1D-CNN, and (c) LSTM architectures, enabling a direct comparison of prediction errors among the evaluated DL models. As illustrated in Figure 7(a), the MLP model demonstrates strong classification capability with a relatively low number of FN, indicating its effectiveness in detecting ransomware traffic. In Figure 7(b), the 1D-CNN model shows slightly higher misclassification rates compared to MLP, particularly in identifying certain attack instances. Similarly, Figure 7(c) shows that the LSTM model also produces several misclassifications, reflecting limitations in capturing meaningful relationships within the tabular feature representation.




(a) (b) (c)

Figure 7. Confusion matrix of DL models: (a) MLP, (b) 1D-CNN, and (c) LSTM

The observed misclassifications mainly occur in samples that exhibit overlapping statistical characteristics between attack and benign traffic flows. This finding highlights the inherent difficulty of separating certain network traffic patterns solely based on aggregated statistical features. Visual inspection shows that the MLP architecture produces the fewest FN, which is particularly important in ransomware detection scenarios. Quantitative performance results are summarized in Table 3.

The lower recall values observed in 1D-CNN and LSTM indicate that these architectures struggle to capture meaningful relationships within the tabular feature representation. Unlike image or time-series datasets, the UNSW-NB15 features are primarily statistical aggregates of network traffic behavior rather than sequential signals.

Table 3. DL model performance

| Model | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| MLP | 0.8859 | 0.8723 | 0.9287 | 0.8996 |
| 1D-CNN | 0.8683 | 0.8690 | 0.8959 | 0.8823 |
| LSTM | 0.8704 | 0.8659 | 0.9046 | 0.8848 |

### 3.3. Comparative discussion ML vs DL

Experimental results demonstrate that ensemble-based ML models consistently outperform DL architectures on decontaminated tabular data. The superior performance of RF (0.9027) compared to MLP (0.8859) highlights the effectiveness of bagging strategies in reducing variance and improving generalization in structured datasets. This finding is consistent with [23], which reported that bootstrap aggregating significantly mitigates overfitting in tree-based models.

Furthermore, the observation that classical ML models can match or exceed DL performance on structured feature representations aligns with [24]. A recent comprehensive review [25], also emphasizes that in ransomware detection, model performance is heavily influenced by data preprocessing quality and the suitability of the chosen architecture rather than model complexity alone. In this context, tree-based models demonstrate robustness against irrelevant or weakly informative features [13], [14], while XGBoost effectively captures complex feature interactions through gradient boosting mechanisms [6].

The relatively higher performance of MLP compared to 1D-CNN and LSTM can be explained by the mismatch between model inductive bias and data characteristics. The UNSW-NB15 dataset consists of statistical aggregation features [15], that lack explicit spatial or temporal dependencies. Consequently, architectures such as CNN and LSTM, which are designed to exploit spatial locality and sequential patterns, do not provide additional learning advantages and may introduce unnecessary model complexity, leading to a potential generalization gap [12]. These findings indicate that increasing architectural complexity does not necessarily lead to improved performance in tabular intrusion detection tasks. Instead, the results reinforce that model effectiveness is highly dependent on the alignment between data structure and learning assumptions. In this study, feature decontamination plays a crucial role by removing misleading attributes, forcing models to learn genuine behavioral patterns rather than exploiting hidden data leakage. This provides a more realistic evaluation compared to many prior studies that report overly optimistic results.

However, it is important to acknowledge several limitations. First, the experiments are conducted on a single dataset (UNSW-NB15), which may limit generalizability across different network environments. Second, although feature decontamination improves evaluation reliability, it may also remove potentially useful high-level statistical indicators, which could affect model performance under certain conditions. Future work should therefore investigate cross-dataset validation and adaptive feature selection strategies to enhance model robustness.

From a practical perspective, the results suggest that lightweight ensemble-based ML models provide an optimal balance between detection accuracy and computational efficiency. This is particularly relevant for real-time deployment in IDS within telecommunication infrastructures, cloud platforms, and IoT environments, where low latency and resource efficiency are critical. The findings also highlight that improving data quality and feature integrity is more impactful than increasing model complexity, offering important guidance for the design of efficient and scalable cybersecurity solutions.

## 4. CONCLUSION

This study has successfully evaluated the performance of ML and DL models for ransomware detection using a rigorous feature decontamination protocol on the UNSW-NB15 dataset. The results demonstrate that ensemble-based ML models systematically outperform more complex DL architectures on decontaminated tabular data. Specifically, the RF algorithm achieved the highest detection integrity with an accuracy of 0.9027 and a recall of 0.9507, which is vital for minimizing undetected ransomware threats. Among DL architectures, the MLP proved the most competitive, achieving an accuracy of 0.8859 and outperforming 1D-CNN and LSTM due to its suitability for aggregated statistical features.

The primary contribution of this research is the validation that feature decontamination prevents overly optimistic results caused by data leakage, thereby providing a more realistic and defensible evaluation

of IDS reliability. The implementation of SMOTE successfully addressed class imbalance, ensuring high detection sensitivity without relying on biased dataset artifacts. These findings highlight that for tabular cybersecurity datasets, ensuring data integrity through feature decontamination can be more influential than increasing model complexity.

From a practical perspective, the results suggest that ensemble-based ML models offer an effective balance between detection accuracy and computational efficiency, making them suitable for deployment in real-time IDS within telecommunication networks and cloud-based infrastructures. Future research may explore cross-dataset validation and the integration of these models into real-time edge computing environments to enhance ransomware detection in large-scale networks.

## FUNDING INFORMATION

The authors state that no funding was used to conduct this research. The researchers' own resources supported this work, and no external grants or financial assistance from any funding agency were received.

## AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

| Name of Author | C | M | So | Va | Fo | I | R | D | O | E | Vi | Su | P | Fu |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sriyanto | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | | | ✓ | |
| Chairani Fauzi | | | ✓ | | ✓ | ✓ | | ✓ | | ✓ | ✓ | | ✓ | |
| Mohd Faizal Abdollah | | | ✓ | | ✓ | | | ✓ | | ✓ | ✓ | | ✓ | |
| Zuriati | ✓ | ✓ | | ✓ | | | ✓ | | | ✓ | | ✓ | ✓ | ✓ |

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

## CONFLICT OF INTEREST STATEMENT

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## DATA AVAILABILITY

The data that support the findings of this study are openly available in the UNSW-NB15 dataset repository provided by the University of New South Wales (UNSW) at https://research.unsw.edu.au/projects/unsw-nb15-dataset. The processed and decontaminated data used in this research are available from the corresponding author upon reasonable request.

## REFERENCES

[1] R. A. M. Alsaidi, W. M. S. Yafooz, H. Alolofi, G. A.-M. Taufiq-Hail, A.-H. M. Emara, and A. Abdel-Wahab, "Ransomware Detection using Machine and Deep Learning Approaches," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, no. 11, 2022, doi: 10.14569/ijacsa.2022.0131112.

[2] A. Raizza and A. Algarni, "Ransomware Detection using Machine Learning: Survey." *Computer Science and Mathematics*, May 23, 2023, doi: 10.20944/preprints202305.1635.v1.

[3] N. Rani, S. V. Dhavale, A. Singh, and A. Mehra, "A Survey on Machine Learning-Based Ransomware Detection," *Advances in Intelligent Systems and Computing*, Springer Singapore, pp. 171–186, 2022, doi: 10.1007/978-981-16-6890-6_13.

[4] I. Emeteveke, O. Adeyeye, and O. Emehin, "The Role of Emerging Technologies in Advancing Edge Computing for Cybersecurity Forensics," *International Journal of Computer Applications Technology and Research*, vol. 13, no. 10, pp. 111–124, 2024, doi: 10.7753/IJCATR1310.1011.

[5] A. Shafique, "Network Intrusion Detection System using UNSW-NB15 Dataset," *Unpublished*, 2024, doi: 10.13140/RG.2.2.34782.50241.

[6] Z. Zoghi and G. Serpen, "Building an intrusion detection system on UNSW-NB15: Reducing the margin of error to deal with data overlap and imbalance," *Concurrency and Computation: Practice and Experience*, vol. 36, no. 25, pp. 1–20, 2024, doi: 10.1002/cpe.8242.

[7] F. J. ALzaher and A. AlJarullah, "Intrusion Detection Using Machine Learning and Deep Learning," *International Journal of*

*Advanced Computer Science and Applications (IJACSA)*, vol. 16, no. 8, 2025, doi: 10.14569/ijacsa.2025.0160844.
[8] M. L. Ali, K. Thakur, S. Schmeelk, J. Debello, and D. Dragos, "Deep Learning vs. Machine Learning for Intrusion Detection in Computer Networks: A Comparative Study," *Applied Sciences*, vol. 15, no. 4, p. 1903, Feb. 2025, doi: 10.3390/app15041903.
[9] N. Thapa, Z. Liu, D. B. KC, B. Gokaraju, and K. Roy, "Comparison of Machine Learning and Deep Learning Models for Network Intrusion Detection Systems," *Future Internet*, vol. 12, no. 10, p. 167, Sep. 2020, doi: 10.3390/fi12100167.
[10] Vinod, A. L. P. Rao, A. Manikandan, and V. Kiranmai, "Comparative Study of Intrusion Detection Systems: Machine Learning Vs Deep Learning Approaches," *Global Journal of Engineering Innovations and Interdisciplinary Research*, vol. 5, no. 5, Oct. 2025, doi: 10.33425/3066-1226.1149.
[11] M. A. Khan *et al.*, "A Deep Learning-Based Intrusion Detection System for MQTT Enabled IoT," *Sensors*, vol. 21, no. 21, p. 7016, Oct. 2021, doi: 10.3390/s21217016.
[12] B. A. Pratomo, M. F. Haykal, H. Studiawan, and D. Purwitasari, "Graph-Structured Network Traffic Modelling For Anomaly-Based Intrusion Detection," *Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI*, vol. 14, no. 2, pp. 240–249, 2025, doi: 10.23887/janapati.v14i2.94959.
[13] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on tabular data?," 2022, *arXiv*, doi: 10.48550/ARXIV.2207.08815.
[14] R. Shwartz-Ziv and A. Armon, "Tabular Data: Deep Learning is Not All You Need," 2021, *arXiv*, *doi*: 10.48550/ARXIV.2106.03253.
[15] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," *2015 Military Communications and Information Systems Conference (MilCIS)*, IEEE, pp. 1–6, Nov. 2015, doi: 10.1109/milcis.2015.7348942.
[16] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Computer Networks*, vol. 174, p. 107247, Jun. 2020, doi: 10.1016/j.comnet.2020.107247.
[17] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a Standard Feature Set for Network Intrusion Detection System Datasets," *Mobile Networks and Applications*, vol. 27, no. 1, pp. 357–370, Nov. 2021, doi: 10.1007/s11036-021-01843-0.
[18] A. Ferriyan, A. H. Thamrin, K. Takeda, and J. Murai, "Generating Network Intrusion Detection Dataset Based on Real and Encrypted Synthetic Attack Traffic," *Applied Sciences*, vol. 11, no. 17, p. 7868, Aug. 2021, doi: 10.3390/app11177868.
[19] S. A. Ajagbe, J. B. Awotunde, and H. Florez, "Intrusion Detection: A Comparison Study of Machine Learning Models Using Unbalanced Dataset," *SN Computer Science*, vol. 5, no. 8, Nov. 2024, doi: 10.1007/s42979-024-03369-0.
[20] R. Tahri, A. Jarrar, A. Lasbahani, and Y. Balouki, "A comparative study of Machine learning Algorithms on the UNSW-NB 15 Dataset," *ITM Web of Conferences*, vol. 48, p. 03002, 2022, doi: 10.1051/itmconf/20224803002.
[21] A. M. Mahfouz, D. Venugopal, and S. G. Shiva, "Comparative Analysis of ML Classifiers for Network Intrusion Detection," *Advances in Intelligent Systems and Computing*, pp. 193–207, 2020, doi: 10.1007/978-981-32-9343-4_16.
[22] D. Fyford, E. Anderson, R. Thomas, and M. Garcia, "Detecting Ransomware through Network Traffic Patterns using Random Forest Machine Learning." *Wiley*, Oct. 17, 2024, doi: 10.22541/au.172918752.27152125/v1.
[23] Z. Zuriati, D. Meilantika, A. Arpan, R. Permata, S. Sriyanto, and Mohd. Z. Mas'ud, "Enhancing Chronic Kidney Disease Classification Using Decision Tree and Bootstrap Aggregating: Uci Dataset Study with Improved Accuracy and Auc-Roc," *Jurnal Teknik Informatika (JUTIF)*, vol. 6, no. 5, pp. 3111–3123, Oct. 2025, doi: 10.52436/1.jutif.2025.6.5.5271.
[24] S. Sriyanto, R. A. Aziz, D. A. Rahayu, Z. Zuriati, M. F. Abdollah, and I. Irianto, "Comparative Analysis of Machine Learning Algorithms for Dengue Fever Prediction Based on Clinical and Laboratory Features," *Jurnal Teknik Informatika (JUTIF)*, vol. 6, no. 6, pp. 5944–5955, Jan. 2026, doi: 10.52436/1.jutif.2025.6.6.5309.
[25] E. Kritika, "A comprehensive literature review on ransomware detection using deep learning," *Cyber Security and Applications*, vol. 3, p. 100078, Dec. 2025, doi: 10.1016/j.csa.2024.100078.

## BIOGRAPHIES OF AUTHORS

**Sriyanto** is an academic staff member at the Department of Informatics, Institute of Informatics and Business Darmajaya, Lampung, Indonesia. He received his Doctoral degree (Ph.D.) from Universiti Teknikal Malaysia Melaka (UTeM), Malaysia. His research interests include machine learning, deep learning, and computer vision, with a focus on data-driven modeling and intelligent decision support systems. He has been actively involved in various research projects in predictive modeling and intelligent data processing across agriculture, education, and information systems. His academic work focuses on bridging advanced computational methods with practical technological applications. In this study, he contributed to the comparative analysis of modeling architectures and the optimization of detection algorithms. He can be contacted at email: sriyanto@darmajaya.ac.id.

**Chairani Fauzi** received her Bachelor's degree in Informatics (S.Kom.) from the Institute of Informatics and Business Darmajaya, Lampung, Indonesia. She obtained her Master of Engineering (M.Eng.) and Doctoral degree (Ph.D.) from Universitas Gadjah Mada (UGM), Yogyakarta, Indonesia. She is currently a senior lecturer and researcher at the Department of Informatics, Institute of Informatics and Business Darmajaya. Her research interests include artificial intelligence, machine learning, and data science. She has a strong publication record indexed in Scopus and SINTA. She can be contacted at email: chairani@darmajaya.ac.id.

**Mohd Faizal Abdollah** is currently a Professor and Senior Lecturer at Universiti Teknikal Malaysia Melaka (UTeM), Malaysia. His research expertise focuses on network security, malware detection, and network management. He led a significant sub project under the CMERP initiative in collaboration with Cyber Security Malaysia, specializing in malware detection, eradication, and mitigation. Additionally, he has been involved in various research grants including fundamental grants for botnet detection and ISIF grants for graph-based botnet analysis. He has authored numerous scientific publications in the cybersecurity field and manages advanced courses in IT security and network management. In this study, he provided expert supervision on the malware analysis framework and validated the experimental results. He can be contacted at email: faizalabdollah@utem.edu.my.

**Zuriati** is a lecturer at the Department of Internet Engineering Technology, Politeknik Negeri Lampung, Lampung, Indonesia. She obtained her Master's degree in Computer Science from IPB University. Her research interests include artificial intelligence, data mining, deep learning, and computer vision, with a focus on intelligent systems and digital transformation for precision agriculture. She has published several scientific articles in national and international journals indexed by SINTA and Scopus. She can be contacted at email: zuriati_mi@polinela.ac.id.