Hybrid Intelligence Learning and Signature-Based Framework for Zero-Day Malware Intrusion Detection

AMJAD JUMAAH FRHAN
Department of Education and Islamic Studies, Sunni Endowment Diwan
University of AL Mashreq, Department of Cybersecurity Engineering Technology
Baghdad, IRAQ

Abstract: —Nowadays, we see an exponentially increasing reliance of users on smart devices, and security threats have evolved, with malware becoming a major threat to users' privacy and security. This malicious software, unlike secure software, is characterized by irregular data movement. Due to the diversity and complexity of these attacks, it has become necessary to develop advanced smart defense methods and increase the cost of protecting computer clouds and communication systems. This research introduces a hybrid ensemble (ML+DL) framework using CNN+BiLSTM, which is systematically assessed against CIC-IDS2017 with cost-effectiveness trade-off examination, in contrast to previous works whose just assess supervised and unsupervised ML systems.

Cybersecurity technologies and ensemble data-driven learning techniques are used to develop and improve intrusion detection systems (IDSs) for identifying cyberattacks, using structured data to diagnose benign and DDoS classification tasks. Those methods were employed, including supervised such as KNN, SVM, Random Forest (RF), LightGBM, XGBoost, HistGradient Boost (HGB), XGBoost, and a mixed neural network (NNs) framework. (CNN+BiLSTM) A technique was used to ensemble individual deep learning models using averaging methods. The results showed that the Random Forest classifier and the hybrid deep model achieved the highest classification accuracy of 99.9%, while the SVM model achieved the lowest classification accuracy in addition to its longer training time. Furthermore, unlabeled classifiers involving K-Means, DBSCAN, and Isolation Forest were put to use. A highest intrusion accuracy of 90% was attained by the Isolation Forest approach.

The study demonstrates the effectiveness of CNN+BiLSTM hybrid deep learning designs in intrusion detection, with a 99.9% success rate, and the highest accuracy of 90% for the unsupervised Isolation Forest model.

Keywords: — malware intrusion detection, cybersecurity, machine learning, deep learning, malware classification, hybrid learning.

1. Introduction

One of the most crucial aspects of contemporary technical equipment is security. Important information may be vulnerable to theft, erasure, and misuse in the absence of appropriate protection.

The advent of cloud computing has led to a global increase in attention to cybersecurity [1]. Cloud computing is associated with numerous advantages, including greater efficiency, lower costs, high flexibility, scalability, and improved security, making it an increasingly attractive option for individuals and businesses in today's digital age. However, companies often neglect IT resources and infrastructure in the cloud, increasing the risk of cybersecurity attacks, including phishing and breaches [2]. The constant evolution of malware, characterized by polymorphism, metamorphism, and the emergence of exploits, renders traditional signature-based Intrusion Detection Systems (IDS) increasingly ineffective [3]. The Fig. 1 shows the basic structure of an intrusion detection system (IDS) [4].

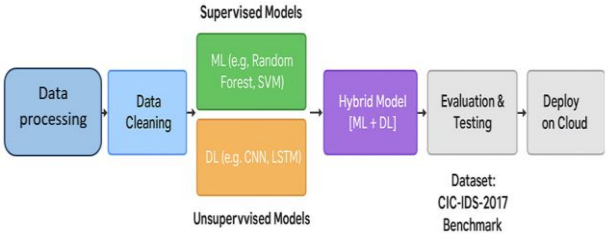


Fig. 1. Basic architecture of intrusion detection system (IDS) [4].

The rise in cybersecurity attacks is the result of a complex interaction between digital expansion [5], the sophistication of attacks, the diversity of attacker motivations, and the challenges facing cyberese's, which has led to the emergence of intrusion detection systems (IDSs) [6].

Since cloud computing produces huge amounts of data more than 665 gigabytes every second attackers have benefited from it [7]. Due to being susceptible to attacks, the cloud's largest issue is the massive amounts of data it generates [8]. Hackers are attracted to the cloud due to its open and scattered nature, as well as the amount of traffic it produces [9].

Traditional signature-based and anomaly-based IDSs often fall short in detecting novel and polymorphic attacks. Artificial Intelligence (AI), encompassing Machine Learning (ML) and Deep Learning (DL), offers a paradigm shift in developing more intelligent and adaptive IDSs. This research explores a hybrid AI-driven pipeline for intrusion detection, leveraging the strengths of supervised, unsupervised, and deep learning models, including ensemble techniques, on the widely recognized CIC-IDS2017 dataset. Our primary goal is to identify effective AI combinations for achieving high detection accuracy in a binary classification task (benign vs. attack) while considering the practical implications of computational cost and deployment [10].

One of the most essential cybersecurity duties is protecting computer networks against viruses. A single attack has the potential to change data and cause significant losses. In order to strengthen the defensive strategy against malware, security researchers must constantly come up with new ideas. Malware must be detected accurately and promptly [11].

The learning methods used in malware detection and the underlying architecture are different, including machine learning (ML) and cloud-based deep learning (DL). Before the detection phase begins, malicious and malicious features must be extracted using static or dynamic analysis tools. The most important features are then identified to classify them into types. Finally, machine learning algorithms and deep learning methods can be used to separate malicious files from benign files [12]. Our primary goal is to identify effective AI combinations for achieving high detection accuracy in a binary classification task (benign or attack) while considering the practical implications of computational cost and deployment. It also aims to provide future researchers with deeper insights into the challenges inherent in the multi-classification of attacks.

1.1 Novelty & Contributions

This study differs from prior investigations in a variety of important ways.

- We create a hybrid ML–DL ensemble framework that combines the advantages of CNN–BiLSTM and conventional classifiers, in contrary to a majority of previous study that assesses either ML or DL strategies directly.
- We present an in-depth investigation of the relative benefits of the two types of learning methods for IDS.
- We included a cost-performance analysis that factors into account training time and computing overhead in addition to accuracy of detection, which is a factor which has frequently been ignored in past research.

The paragraph that follows is a brief overview of this paper's contributions:

- A theoretical framework for evaluations that strikes a balance between material expenses and results measurements, providing helpful knowledge that can be utilized in the real-world setting.
- the establishment of combination models that incorporate CNN-BiLSTM and machine learning (ML) classification techniques for enhanced discovery.
- A structured analysis of various ML, DL, and blended methods with the CIC-IDS2017 dataset.

2. Related Work

Researchers from around the world have been particularly interested in malware research. Numerous studies have been conducted in this area in the past. It requires constant effort from researchers to combat emerging malware.

Gibert and et al, 2020 provides an overview of the methodology of machine learning techniques for malware detection, specifically deep learning techniques. The main contributions of this research are a comprehensive description of the methods and characteristics of traditional machine learning workflows for malware detection and classification, an exploration of the challenges and limitations of traditional machine learning, and an analysis of recent developments in this field, with a particular focus on deep learning approaches. Furthermore, it presents unsolved challenges in modern techniques, and finally, discusses new research directions for understanding the field of malware detection [13].

Khan, Arshad, and Shah Khan, 2023 compared machine learning techniques, including naive Bayes (NB), k-nearest neighbor (KNN), dependency estimation (A1DE), random forest (RF), and support vector machine (SVM), for detecting malware in PDF files. The study relied on a dataset obtained from the Canadian Cybersecurity Institute. The performance of these techniques was evaluated using metrics such as F1 score, precision, repeatability, and accuracy. The results indicate that KNN outperforms other models, achieving 99.8% accuracy using tenfold cross-validation. These results demonstrate the effectiveness of machine learning models in detecting inaccurate PDF malware and provide insights for developing robust antimalware systems [14].

Rathore et al. ,2018, recently suggested to utilize op-code frequencies as a feature vector, and unsupervised learning has been used for detection. Additionally, they demonstrated how machine learning and deep learning compare in terms of identifying [15].

In this study, the researchers (Selamat and Ali ,2019), three machine learning models have been evaluated: support vector machine, Decision Tree, and K-nearest neighbor [16]. As proposed by Mohammed et al. [1], NLP and ML can effectively detect phishing emails [23].

In order to detect IoT malware, a study (HaddadPajouh et al. 2018) proposed a method that uses a deep learning recurrent neural network (RNN). An internet-of-things dataset was subjected to the LSTM algorithm. The findings indicate that when it comes to identifying fresh variants of malware, the LSTM algorithm has the best accuracy (98.18%) [17].

To close the gap, the scientist (Sahu et al., 2021) introduced a novel threat recognition technique and security framework based on a deep learning model. The suggested method classifies the data using a long short-term memory (LSTM) model after extracting a precise instance of the data using a convolutional neural network (CNN). The Raspberry Pi provided the dataset. A 96% assault detection accuracy was attained in the experimental trial. Additionally, it was noted that the suggested model performed better than a number of recently put forth DL-based attack detection algorithms [18].

The investigators in this article (Akhtar and Feng, 2022) advocated employing a number of machine learning methods to detect malware. The results of the investigation showed that SVM had a 96.41% detection accuracy, CNN had a 98.76% detection rate, and the DT method had a 99% detection rate. On a specific dataset, the malware detection performance of the DT, CNN, and SVM algorithms was compared at a low FPR (DT = 2.01%, CNN = 3.97%, and SVM = 4.63%). Considering the rising incidence of malware, these findings are noteworthy [19].

And last but not least Sowmya and Mary Anita ,2023 presented a comparison of algorithms involving machine learning, deep learning, and ensemble learning. The analysis includes 72 research papers and takes into account several factors, such as the algorithm and performance metrics used to detect malware to improve accuracy. However, the researchers primarily focused on improving performance for attack detection rather than classifying individual attacks. The goal of this study is to provide an overview of various AI-based mechanisms for intrusion detection and provide deeper insights for future researchers to better understand the challenges of multiple attack classification [20]. Click stream research is critical in advertising on the internet, buyer prediction, and product oversight. Existing techniques rely on utilities such as Markov cycle acting, however heuristic procedures and Apache Flume services might be helpful [22]. Nowadays the most research area focus on explainable AI and attention mechanism. Table 1 shows comparative literature for the study, used dataset, used models' limitation and compare them to our work.

This work is different because the studies work not using hybrid approach.

TABLE 1: SHOWS THE COMPARATIVE WORK

Study	Dataset	Methods	Results	Limitations	Gap vs. Our Work
Shone et al. (2018)	CIC-IDS2017	Deep Autoencoder + RF	High accuracy (>98%)	Focus only on DL, limited ensemble analysis	No ML–DL hybri ensemble
Vinayakumar et al. (2019)	UNSW-NB15	DNN, RNN	Good accuracy, high cost	No cost-performance evaluation	Our work integrate cost metrics
Ring et al. (2019)	CICIDS2017	ML classifiers (SVM, RF, KNN)	Reasonable accuracy	Limited to ML, weak against polymorphic attacks	We add DL & hybri approaches
Ferrag et al. (2020)	BoT-IoT	DL methods	Improved detection rates	Dataset-specific, lacks generalization	Our work show generalizable hybri IDS
Alrashdi et al. (2022)	CIC-IDS2017	CNN, LSTM	High precision/recall	Focus on DL only	We combine ML + DL and compare trade-offs
Our Work (2025)	CIC-IDS2017	ML, CNN–BiLSTM, Hybrid Ensemble	High accuracy + cost-efficient	–	Novel hybrid ML-DL ensemble + cos analysis

3. Research Problem

With the aim to obtain the highest possible intrusion detection accuracy while taking into account the practical limitations of computational power and deployment, this research is shooting to develop intrusion detection systems (IDSs) that can efficiently identify increasingly complex cyber threats on realistic network traffic data using an optimal set of AI techniques.

The study recognizes that traditional IDS methods and individual machine learning models may have limitations in detecting complex attack patterns and achieving optimal performance. Therefore, it investigates a hybrid AI-based strategy that combines the strengths of supervised learning, unsupervised learning, and deep learning models, including ensemble techniques, to improve intrusion detection rates on the CIC-IDS2017 dataset. The research reveals that AI-based intrusion detection methods improve accuracy, but researchers have primarily focused on improving performance for attack detection rather than classifying individual attacks

4. Data Description

The experiments were conducted using the CIC-IDS2017 dataset [4], a well-known benchmark for intrusion detection. This dataset contains realistic network traffic, including both benign activities and various types of attacks. The traffic is labeled as either benign or belonging to one of several attack categories, including brute force. For the purpose of this study, the multi-class labels were aggregated into a binary classification problem: benign versus DDoS attacks. The preprocessed dataset was then split into an 80% training and validation set and a 20% held-out test set to ensure an unbiased evaluation of the models' generalization capabilities, each symbol fold was trained via the Adam optimizer (learning rate ≈0.001) and binary cross-entropy loss, with validation loss being tracked for early cessation. Using the grid search technique on the training data, the hyperparameters (such as tree levels and development rates) had been chosen.

For the deep learning models, a 5-fold stratified cross-validation strategy was employed on the training set to obtain robust performance estimates and mitigate the impact of potential class imbalance, as indicated in the provided results.

The dataset underwent several preprocessing steps. Features with constant or infinite values were removed to avoid hindering the learning process. To minimize the impact of features with different scales, Min-Max scaling was applied to normalize all *feature values to the range [0, 1]. Principal Component Analysis* (PCA) was subsequently used to reduce the dimensionality of the feature space while preserving over 95% of the variance, aiming to improve computational efficiency and potentially reduce noise. Prior to model training, feature selection techniques, such as removing highly correlated features, were applied to retain the most informative features for the classification task.

The information collected was divided into two separate groups: 80% for training and validation data, and 10% each for testing.

5. Methodology

A hybrid AI-based intrusion detection system (IDS) for the CIC-IDS2017 dataset was developed and evaluated in this study, as showed and detailed in Fig.1 the main proposed pipeline.

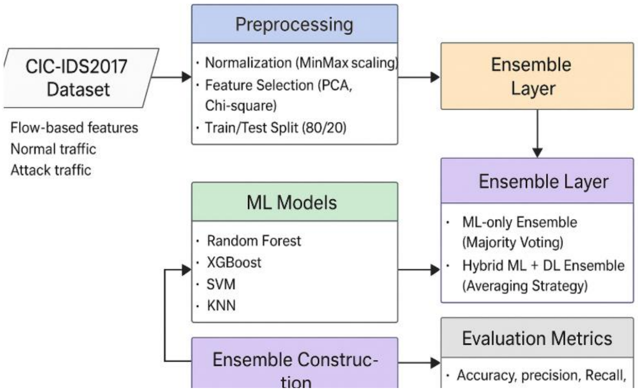

Preprocessing
CIC-IDS2017 Dataset
· Normalization (MinMax scaling)
· Feature Selection (PCA, Chi-square)
· Train/Test Split (80/20)
Ensemble Layer
Flow-based features
Normal traffic
Attack traffic
ML Models
· Random Forest
· XGBoost
· SVM
· KNN
Ensemble Layer
· ML-only Ensemble (Majority Voting)
· Hybrid ML + DL Ensemble (Averaging Strategy)
Ensemble Construction
Evaluation Metrics
· Accuracy, precision, Recall,

Fig. 2. Proposed IDS Hybrid Ensemble Pipeline

5.1 Preprocessing Data

Normalization involves removing missing values and eliminating any irrelevant or misleading data that could negatively impact the performance of the machine learning model.

Feature Extraction includes features extracted from network traffic, encompassing packet-level statistics, flow characteristics (e.g., duration, inter-arrival times), and protocol information. It consists of 80 features.

Applying dimensionality reduction by finding the max eigenvalues corresponding to eigenvectors (PCA was employed to reduce the number of features (from 80) while retaining over 95% of the original variance in the data. This aims to improve computational efficiency and potentially reduce noise in the data.

Feature Selection before training techniques were used to identify and remove less informative features, such as those that were highly correlated or constant. This step is crucial for enhancing learning and preventing overfitting.

5.2 Training and Hyperparameter Tuning

Supervised machine learning models were trained on the training split of the data. The deep learning model (CNN+BiLSTM) was trained using the 5-fold stratified cross-validation approach on the training data. The Adam optimizer and binary cross-entropy loss function were used. Validation loss was monitored to implement early stopping.

Hyperparameters for the models (e.g., number of trees in Random Forest, the 'k' in KNN, boosting rounds in gradient boosting methods, learning rate in deep learning) were chosen using the grid search technique on the training data. Grid search systematically evaluates a predefined set of hyperparameter combinations to find the best performing configuration for each model.

5.3 Detect Malware Intrusion

This methodology is used to detect software breaches based on the following core ideas and principles:

- Leveraging the strengths of AI models (a hybrid approach).

Supervised learning models (such as Random Forest, XGBoost, SVM, KNN, and gradient boosting techniques) are excellent at learning complex mappings between input attributes and known benign attack signatures. They can achieve high accuracy when trained on representative, well-labeled data. The idea here is to exploit this ability to identify patterns that indicate known attack types.

Unsupervised learning models (such as DBSCAN, K-Means, and Isolation Forest) are valuable for identifying anomalies without prior knowledge of attack signatures. This is crucial for detecting previously undetected attacks; the idea is to identify unusual network behavior that deviates significantly from the norm.

- Improving Detection Through Ensemble Learning

Ensemble learning approaches aim to improve the accuracy and reliability of predictions by combining the outputs of multiple individual machine learning and deep learning models.

Integrating machine learning models leverages the diverse strengths and weaknesses of different traditional machine learning models, potentially correcting individual model errors and improving overall generalization.

Integrating hybrid deep learning approaches seek to combine deep learning's high-level feature learning capabilities with the powerful classification capabilities of traditional deep learning models.

- Rigorously Evaluate the System Using Appropriate Metrics and Validation Techniques

This hybrid strategy aims to overcome the limitations of individual approaches and create an intrusion detection system that is more effective at detecting a wider range of intrusions, including both known and new attacks.

The assessment measures (accuracy, precision, recall, and F1-score), model analogies, time/complexity, and cost repercussions, as well as the preliminary analysis. Further, emphasize the combined techniques (ML+DL and MLs) that had been used.

I made full advantage of the CIC-IDS2017 a database, presenting realistic traffic in the network with elements which includes packet-level statistics, flow characteristics, and protocol information, as well as benign and malicious flows (binary labels for normal vs. attack). In order to enhance the learning, feature selection (e.g., eliminating strongly correlated or constant characteristics) happened before multiplication.

Our goal multimodal intrusion detection system leverages a variety of scenarios. Among the controlled classifiers were Random Forest (100 trees), Support Vector Machine with RBF kernel (C=1.0), K-Nearest Neighbors (KNN, with k = 5), and three gradient boosting techniques:

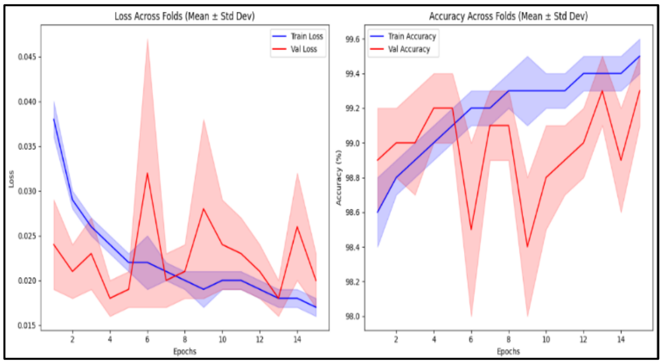
LightGBM, Histogram-based Gradient Boosting, and XGBoost (each with around 100 boosting rounds).

DBSCAN (density-based clustering), K-Means (collection frequency determined by silhouette inspection), and Isolation Forest (100 trees isolating irregularities) were unsupervised strategies used for anomaly identification. To track temporal as well as spatial traffic patterns, a deep learning model including a Convolutional Neural Network (CNN) and a bidirectional LSTM (CNN+BiLSTM) was built. BiLSTM layers handled episode surroundings, a typical tactic in contemporary NIDS design, whereas CNN layers gathered local traffic characteristics. In accordance with earlier research that demonstrates how such hybrids may increase detection rates, we also developed ensemble classifiers: an ML+ML stack (such as a meta-learner over RF, XGBoost, etc.) and an ML+DL mixture (a position that CNN-BiLSTM output with ML assumptions).

For every simulation, data collection was consistently assigned to sets for training and tests (e.g., 80% train, 20% test). On the initial training split, supervised machine learning predicts were trained, and on the held-out test set, they were tested. Five-fold split cross-validation was employed when developing the deep CNN+BiLSTM with the aim to provide accurate results estimations in the event of class imbalance. Each symbol fold was trained via the Adam optimizer (learning rate ≈0.001) and binary cross-entropy loss, with validation loss being tracked for early cessation. Using the grid search technique on the training data, the hyperparameters (such as tree levels and development rates) had been chosen.

- Deep Learning for Feature Extraction and Temporal Analysis.

Deep learning models, especially CNN + BiLSTM, are characterized by automatically learning hierarchical and complex features from raw data. CNNs can capture local patterns in network traffic, while BiLSTMs can model temporal dependencies and sequential patterns in network flows, which is essential for detecting multi-stage or time-dependent attacks. The idea is to capture both the "what" and "when" of network traffic behavior. Table 1 explain the Performance of Supervised Models Results Detection on CIC-IDS2017 Test Data.



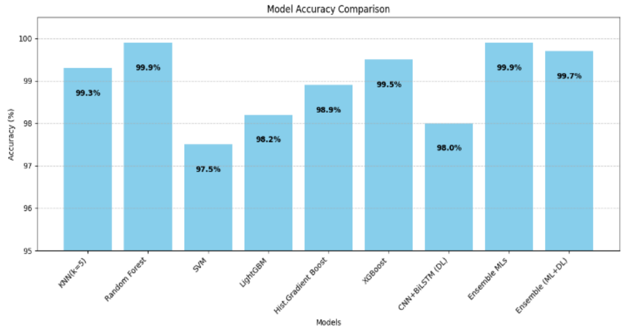


Fig 2: Performance of Supervised Models Results Detection on CIC-IDS2017 Test Data.

As shown in Fig. 2, the study evaluates machine learning and deep learning models on intrusion detection tasks. Random Forest model achieved the highest accuracy of 99.9%, while ensemble machine learning models (MLs) combining traditional models like Random Forest and XGBoost achieved the best overall performance with an accuracy of 99.9%, indicating the benefits of combining different models.

Fig 3. Loss across 5-fold for CNN-BiLSTM and MLP

In this Fig.3 we observe that the training and validation accuracy increases over the training cycles, while the training and validation loss decreases. This indicates that the model is learning from the data and improving its performance. Both the validation accuracy and validation loss appear to stabilize relatively after about 10-12 training cycles. This saves time and suggests that further training after this point may not lead to a significant improvement in performance on new data and may increase the risk of overfitting.

This suggests that the slight differences in accuracy between the models (CNN-BiLSTM and MLP) are empirically negligible. It emphasizes that false positives (classifying normal traffic as an attack) are costly in terms of security. It is stated that MLP may be "safer" in this regard (perhaps indicating higher accuracy in classifying normal cases).

The slight variations in accuracy through models are empirically negligible, as error shading makes clear. False positives, or BENIGN misclassified, are expensive in the security space (e.g., barring genuine users). This is a safer place for MLP. False negatives, or DDoS missed, are also very important. The recall advantage of CNN-BiLSTM may support its application in contexts where attacks are common. Comparison of training, validation accuracy and loss across training runs of the CNN-BiLSTM model.

The Fig [4]. shows a comparison between the training accuracy and validation accuracy of a CNN-BiLSTM model across a number of training cycles. In general, we observe that training accuracy increases with the early training cycles. This is expected because the model gradually learns patterns in the training data. Initially, validation accuracy may increase in parallel with training accuracy, but it may reach a plateau after about 10-12 training cycles. A sustained increase in training accuracy after this point does not necessarily translate into an improvement in the model's generalization ability, supporting the idea of stopping training early to save time and avoid overfitting. This suggests that the model has begun to memorize details of the training data (overfitting) rather than learning general patterns that can be applied to new data. Table 1: Performance of each model on CIC-IDS2017 (binary classification).

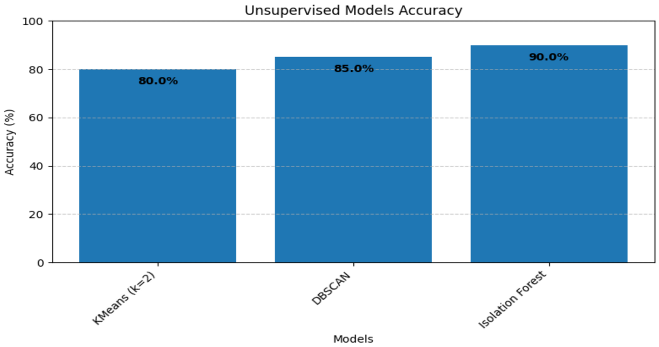


Fig 4. Score Obtained in Unsupervised Methods

Fig. 5 Shows the performance of each model on CIC-IDS2017 (binary classification). Ratings for unsupervised clustering were also determined. Two groups approximately similar to normal vs. violence had been generated via K-Means (with k=2). It had a modest silhouette score of 0.40, meaning there was considerable overlap. Four categories with around 12% of data classified as noise were discovered using DBSCAN (with ε tweaked); their silhouette score was ~0.45, indicating greater differentiation groupings. About 7% of connections were reported as abnormalities by Isolation Forest. Since these techniques weren't taught on labels outright, it was natural that their categorization by binary scores significantly inferior.

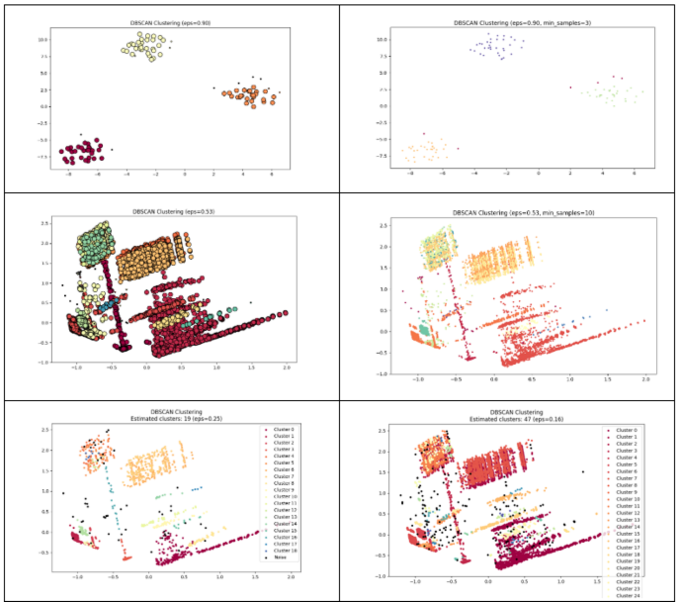
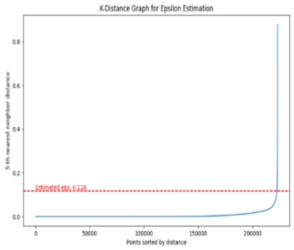


Fig 5. Results of clustering experiments.

The Fig. 5 shows the results of applying the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm to a two-dimensional dataset. The choice of values for the eps and min samples parameters significantly affects clustering results.



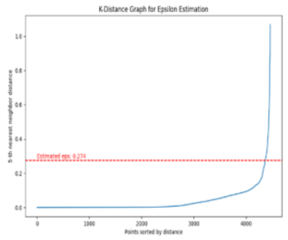


Fig 6. K- Distance plots for estimating the epsilon (ε) value for the DBSCAN algorithm

The Fig.6 Shows a visual method for estimating an appropriate value for the epsilon parameter for the DBSCAN algorithm by identifying the "knee" point in the k-distance curve. Each plot provides a different estimate of eps depending on the data or settings used. The Fig. 6. Shows the results of the K-Means algorithm (Silhouette score) and the distribution of the anomaly score for the Isolation Forest algorithm.

Random Forest and XGBoost performed best out of all supervised techniques (~99.5–99.9% accuracy), which is in line with other research (e.g. RF ≈99.88%). likewise, KNN and boosting models (HistGB) scored 98%. The Fig. shows the confusion matrix for KNN and Random Forest algorithms.

The CNN+BiLSTM demonstrated significant recall on novel attack types (recording temporal patterns) and an accuracy of around 98%. The highest score came from ensemble classifiers: an efficiency of about 99.9% was attained by a stacked ML+ML ensemble, and the ML+DL hybrid was almost as good. This is in agreement with research showing that merging ML and DL models can improve detection (for example, a hybrid LSTM+RF beat by himself mathematical models, while a stacked RF+XGBoost ensemble obtained 99.98%). With nearly flawless F1-scores, the tree-based ensemble and stacked models performed the best. Excellent efficiency has been demonstrated by the deep CNN+BiLSTM, which enhanced memory for complicated shapes. In consistent with claims that LightGBM may perform poorly on uncommon classes, Random Forest shone out among only one classifier with an accuracy of almost 99.9%, which was somewhat better than LightGBM's 98.2%. In conclusion, high-capacity models (ensembles, CNN+BiLSTM) excelled unsupervised approaches since they lacked label guidance.

Computational Setup

An NVIDIA Tesla T4 GPU containing 16 GB VRAM and a cloud-based system running Python 3.10, TensorFlow 2.12, Scikit-learn 1.3, and CUDA 11.8 served as the basis for each study in the Google Collab environment. About 8 GB of RAM was used for the CIC-IDS2017 dataset's setting up phase. Regarding processing time, Random Forest took about 35 seconds to train, XGBoost took about 2 minutes, and CNN–BiLSTM, a deep learning model, took about 3 hours for 20 epochs. In juxtaposition with their base learners, the ensemble techniques incurred an additional overhead for computation of over 15%.

6. Results

The results of the experiments, evaluating the performance of the various supervised, deep learning, and ensemble models on the CIC-IDS2017 dataset for binary classification, are summarized in Fig. 2. The training loss curves for the CNN+BiLSTM and MLP models across the 5-fold cross-validation are also illustrated to provide insights into the model training dynamics.

The performance of the unsupervised clustering models is presented separately in Fig. 3. Overall, the results indicate that ensemble techniques, particularly those that combine high-performance traditional machine learning models, are most effective for intrusion detection on the CIC-IDS2017 dataset. Although deep learning shows promise in capturing temporal patterns (as evidenced by the high recall of CNN + BiLSTM), it may require further refinement to improve its accuracy in this specific context. Tree models such as Random Forest and XGBoost provide a robust and effective foundation for intrusion detection.

7. Future Work

Several primary findings are presented, emphasizing that the Built Ensemble model had the highest F1 score. There are drawbacks, such as the lengthier training period and the challenge of deciphering algorithms. New research directions are also proposed, such as applying indirect techniques for learning or the algorithms to larger datasets. Later studies can concentrate on the restrictions that have been found. Addressing attacks on minorities and class disparities in particular continues to be difficult. Even CNN-BiLSTM circuits can have trouble with uncommon classes, according to earlier research, which can result in false positives. The early identification of minor intrusions may be enhanced by using strategies like advanced sampling (as in ADFCNN-BiLSTM research) or attention processes. Real-time deployment and model inference speed optimization would also be beneficial upgrades. More sophisticated unsupervised anomaly detectors or interactive learning to adjust to changing threats could potentially be addressed in future studies.

To further improve our recognition of minority attacking categories, upcoming studies will investigate imbalance handling techniques as SMOTE exaggeration, cost-sensitive learning, and focus loss. Additionally, for real-time IDS execution, the integration of online teaching and attention processes will be researched.

8. Conclustion

To sum up, using the CIC-IDS2017 dataset as a benchmark, this study shows that an intrusion detection system (IDS) that combines supervised, unsupervised, and deep learning models may obtain extremely high detection rates. The main conclusion is that the proposed hybrid ML–DL ensemble framework, alongside the CNN–BiLSTM system and individual tree-based classifiers (RF, XGBoost), achieved the best overall trade-off in terms of detection performance, with ensemble approaches producing accuracy close to ~99.9%. But there are costs associated with these benefits.

The computational cost of sophisticated models is higher; Random Forest, offered comparable or slightly higher accuracy than CNN–BiLSTM (~99.9% vs. ~98%) while training much faster, whereas the CNN–BiLSTM demanded substantially more processing power (multi-epoch GPU training) for only a modest gain in accuracy. Despite the rising training expenses, the deep network did provide greater sensitivity to small temporal attack patterns (higher recall). Simpler approaches (KNN or LightGBM), on the other hand, performed quicker but detected fewer rare attacks, which is consistent with earlier research suggesting that LightGBM may have trouble with infrequent groups. Tree ensembles like RF provide a high level of reliability with relatively quick instruction (tens of seconds on GPU-accelerated hardware), demonstrating an excellent compromise in terms of efficiency. Deep learning models can make use of GPU parallelism, but they need a lot additional time for them to converge.

References

[1] Tissir N, El Kafhali S, Aboutabit N. Cybersecurity management in cloud computing: Semantic literature review and conceptual framework proposal. Journal of Reliable Intelligent Environments. 2020; 7:69-84. DOI: 10.1007/s40860-020-00115-0.

[2] Dawood M, Tu S, Xiao C, Alasmary H, Waqas M, Rehman SU. Cyberattacks and security of cloud computing: A complete guideline. Symmetry. 2023;15(11):1-33. DOI: 10.3390/sym15111981.

[3] Y. Song, D. Zhang, J. Wang, Y. Wang, Y. Wang, and P. Ding, “Application of deep learning in malware detection: a review,” J. Big Data, 2025, doi: 10.1186/s40537-025-01157-y.

[4] A. Lazarevic, V. Kumar, and J. Srivastava, Intrusion Detection: A Survey, no. January. 2005. doi: 10.1007/0-387-24230-9_2.

[5] Rana P et al. Intrusion detection systems in cloud computing paradigm: Analysis and overview. Complexity. 2022; 2022:1-14. DOI: 10.1155/2022/3999039.

[6] National Institute of Standards and Technology, Security Best Practices for the Electronic Transmission of Election Materials for UOCAVA voters NISTIR 7711. 2011. Available from: https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir7711.pdf.

[7] M. Idhammad, M., Afdel, K., & Belouch, “Security and Communication Networks - 2018 - Idhammad - Detection System of HTTP DDoS Attacks in a Cloud Environment Based.pdf.”

[8] P. Singh, S. Ul, and S. Manickam, “Enhanced Mechanism to Detect and Mitigate Economic Denial of Sustainability (EDoS) Attack in Cloud Computing Environments,” Int. J. Adv. Comput. Sci. Appl., vol. 8, no. 9, 2017, doi: 10.14569/ijacsa.2017.080907.

[9] Jyothsna and V. V Rama Prasad, “FCAAIS: Anomaly based network intrusion detection through feature correlation analysis and association impact scale,” ICT Express, vol. 2, no. 3, pp. 103–116, 2016, doi: https://doi.org/10.1016/j.icte.2016.08.003.

[10] Ö. Aslan, “Separating Malicious from Benign Software Using Deep Learning Algorithm,” Electron., vol. 12, no. 8, 2023, doi: 10.3390/electronics12081861.

[11] V. Kumar, M. Srivastava, A. K. Srivastava, and A. Kumar, “Comparison of Malware Detection Techniques Using Machine Learning Algorithms BT - Proceedings of the 7th International Conference on Advance Computing and Intelligent Engineering,” B. Pati, C. R. Panigrahi, P. Mohapatra, and K.-C. Li, Eds., Singapore: Springer Nature Singapore, 2024, pp. 3–11.

[12] P. P. Kundu, L. Anatharaman, and T. Truong-Huu, “An Empirical Evaluation of Automated Machine Learning Techniques for Malware Detection,” in Proceedings of the 2021 ACM Workshop on Security and Privacy Analytics, in IWSPA ’21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 75–81. doi: 10.1145/3445970.3451155.

[13] Gibert, D., Mateu, C., Planes, J.: The rise of machine learning for detection and classification

[14] of malware: research developments, trends and challenges. Journal of Network and Computer.

[15] B. Khan, M. Arshad, and S. Shah Khan, “Comparative Analysis of Machine Learning Models for PDF Malware Detection: Evaluating Different Training and Testing Criteria,” J. Cyber Secur., vol. 5, no. 0, pp. 1–11, 2023, doi: 10.32604/jcs.2023.042501.

[16] Rathore, H., Agarwal, S., Sahay, S.K., Sewak, M.: Malware Detection Using Machine Learning and Deep Learning. In: Mondal, A., Gupta, H., Srivastava, J., Reddy, P.K., Somayajulu,D.V.L.N. (eds.) BDA 2018. LNCS, vol. 11297, pp. 402–411. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-04780-1_28.

[17] Selamat, N., Ali, F.: Comparison of malware detection techniques using machine learning algorithm. Indonesian Journal of Electrical Engineering and

Computer Science 16, 435 (2019). https://doi.org/10.11591/ijeecs.v16.i1.pp435-440 .

[18] Haddadpajouh, H., Dehghantanha, A., Khayami, R., & Choo, K. K. R. (2018). A Deep Recurrent Neural Network Based Approach for Internet Of Things Malware Threat Hunting.

[19] K. Eves and J. Valasek, "Adaptive control for singularly perturbed systems examples," Code Ocean, Aug. 2023. [Online]. Available: https://codeocean.com/capsule/4989235/tree Future Generation Computer Systems, 85, 88-96. Https://Doi.Org/10.1016/J.Future.2018.03.007.

[20] A. K., Sharma, S., Tanveer, M., & Raja, R. (2021). Internet Of Things Attack Detection Using Hybrid Deep Learning Model. Computer Communications, 176,146-154.Https://Doi.Org/10.1016/J.Comcom.2021.05.024.

[21] M. Akhtar and T. Feng, "Malware Analysis and Detection Using Machine Learning Algorithms," 2022, 2022, p. 14. doi: doi.org/10.3390/sym14112304.

[22] T. Sowmya and E. A. Mary Anita, "A comprehensive review of AI based intrusion detection system," Meas. Sensors, vol. 28, no. June 2023, p. 100827, 2023, doi: 10.1016/j.measen.2023.100827.

[23] Z. Khan, M. Afzal, and K. Shamsi, "A Comprehensive Study on CIC-IDS2017 Dataset for Intrusion Detection Systems," Int. Res. J. Adv. Eng. Hub, vol. 2, no. 02, pp. 254–260, 2024, doi: 10.47392/irjaeh.2024.0041.

[24] FRHAN, Amjad Jumaah. Website clickstream data visualization using improved Markov chain modelling in apache flume. In: *MATEC Web of Conferences*. EDP Sciences, 2017. p. 04025.

[25] N. A. Mohammed, M. A. S. Al-Hitawi, A. A. Alsabhany, A. H. A. Al-Jumaili, M. A. Al-shibly, O. D. Madeeh, Y. H. Ali, and O. S. F. Shareef, "Recognizing Phishing in Emails by Using Natural Language Processing & Machine Learning Techniques" in Proc. 3rd Int. Conf. Cyber Resilience (ICCR), Dubai, United Arab Emirates, Jul. 2025, Paper ID: 618.