Data Driven Approaches to Cybersecurity using Deep Learning

Shreyas Anand
School of Computer Science and Engineering,
Vellore Institute of Technology, Vellore, Tamil Nadu, India

ABSTRACT
The increasing complexity of cyber threats necessitates innovative approaches for pre-emptive defense mechanisms. This research paper focuses on the application of deep learning techniques as a critical tool for monitoring and preventing cyber-attacks. In the realm of cybersecurity, machine learning, and deep learning classification algorithms play pivotal roles in identifying system irregularities indicative of ongoing attacks. Six classification techniques were employed in this study, including traditional machine learning algorithms (Decision Tree, Random Forest, and Gradient Boosting) and advanced deep learning algorithms (Convolutional Neural Network [CNN], Long Short-Term Memory [LSTM], and LSTM plus CNN). Using metrics such as precision, accuracy, F1-score, and recall, their performance was evaluated on the widely used Kaggle dataset CSIC 2010 Web Application Attacks. The findings of the study reveal that the LSTM with CNN exhibits superior performance, showcasing its effectiveness in detecting and defending against diverse cyber threats. This study underscores the urgency and practical benefits of integrating deep learning into cybersecurity protocols to safeguard networks from external and internal threats.

Keywords
Deep Learning, Cybersecurity, Long Short-Term Memory, Decision Tree, Random Forest, Convolutional Neural Network

1. INTRODUCTION
Cybersecurity plays a pivotal role in protecting computers, servers, and networks from unauthorized access, theft, and disruptions. In today's world, where people are heavily dependent on technologies like the Internet, wireless connections, and smart devices, cybersecurity has become a major concern of the twenty-first century [1]. It is a complex challenge that involves ensuring that systems operate reliably, data stays intact, and sensitive information remains secure [2]. The increasing reliance on digital technologies and the Internet of Things (IoT) has heightened the complexity of cybersecurity challenges. With the rise of cyber threats, cyber-attacks have become more sophisticated, involving attempts to gain unauthorized access to computer systems or compromise user information [3]. These attacks can be financially motivated or aimed at disrupting operations. Some attacks are even orchestrated by state-sponsored actors, making them more complex and capable of affecting global entities [4]. The 2017 NotPetya outbreak is a notable example, originating from a cyber-attack on Ukrainian banks and causing widespread damage globally [2, 3, 4].

This paper explores the application of deep learning, a subset of machine learning, in addressing cybersecurity challenges. By employing artificial neural networks to simulate human cognitive processes, deep learning enhances the ability of cybersecurity systems to recognize and adapt to evolving attack patterns swiftly. This research paper employs various classification algorithms, including machine learning and deep learning, on a relevant network intrusion dataset. The study aims to evaluate the effectiveness of these methods in distinguishing between typical and atypical network assaults.

Key findings indicate that deep learning, when applied effectively, can enhance cybersecurity by improving response times, mitigating risks, and optimizing resource allocation. However, the success of these approaches hinges on the accuracy of the data used for machine learning.

2. RELATED WORK
The realm of cybersecurity has undergone extensive exploration, particularly concerning the integration of deep learning techniques. The surge in studies leveraging machine learning algorithms to detect and mitigate diverse cyber threats is a testament to the dynamic nature of this field. Researchers have utilized both real-time datasets and established datasets from various sources, shedding light on the efficacy of these approaches. A comprehensive understanding of the existing body of work not only offers valuable insights into the current state of cybersecurity but also helps identify gaps in knowledge, setting the stage for further advancements.

One significant contribution comes from Kim et al. [5], who introduced a pioneering approach incorporating Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM) and Deep Neural Network (DNN) techniques. Their focus was on classifying instances as normal or abnormal, utilizing the HTTP DATASET CSIC 2010, a dataset comprising 61,065 instances. Impressively, their evaluation demonstrated an outstanding accuracy of 91.54%, showcasing the potential effectiveness of this hybrid model in cybersecurity applications.

Expanding on the application of deep neural networks (DNNs), Kang et al. devised a DNN-based Intrusion Detection System (IDS) to enhance network safety. The parameters constituting the DNN structure were trained using feature vectors derived from network packets. The DNN calculated probability values for different classes, enabling the model to differentiate between normal and attack packages. This approach significantly contributed to the advancement of network security through the application of deep learning.

Vartouni et al. [6] introduced a novel approach by employing Stacked AutoEncoder on the HTTP DATASET CSIC 2010. The model achieved a notable accuracy of 88.32%, showcasing the efficacy of this distinctive algorithm in anomaly detection.

Additionally, Betarte et al. [7] employed three distinct machine learning techniques—Random Forest, K-Nearest Neighbors (K-NN), and Support Vector Machine (SVM)—on the same HTTP DATASET CSIC 2010. Notably, Random Forest outperformed other methods, achieving an accuracy of 91.54%, emphasizing the significance of algorithm selection in cybersecurity applications.

Tuan et al. [8] extended their study to the UNSW-NB15 dataset, encompassing various network assault types. With machine learning algorithms such as SVM, ANN, Naive Bayes (NB), and Unsupervised Learning (USML), they showcased the high precision of USML at 94.78%. Anwer et al. [9] turned their focus to detecting malicious network traffic using four machine-learning techniques on the well-known NSL-KDD dataset. Their comprehensive evaluation, covering accuracy, specificity, training time, and prediction time, identified Random Forest as the top performer with an accuracy of 85.34%.

Su et al. [10] introduced the BAT model, a deep learning technique designed for identifying hostile network infiltration. Leveraging the NSL-KDD dataset, they achieved an intrusion detection accuracy of 84.25%. Xu et al. [11] proposed a five-layer autoencoder model to enhance the detection of network anomalies on the NSL KDD dataset, achieving an accuracy of 90.61%. Kavitha et al. [12] presented a One-Class Support Vector Machine (OCSVM) approach on NSL-KDD, achieving 81.29% accuracy in intrusion detection.

Ferriyan et al. [13] made significant strides by developing multiple machine-learning models for detecting cyberattacks on the ALLFLOWMETER HIKARI2021 dataset. The dataset includes six types of attacks represented among 555,278 instances and 86 features. Employing KNN, SVM, RF, and MultiLayer Perceptron (MLP) models, they demonstrated an outstanding detection accuracy of approximately 99%.

These diverse studies collectively highlight the versatility and efficacy of machine learning and deep learning techniques in addressing cybersecurity challenges, providing valuable insights that inform the approach to vulnerability testing in this study using CSIC 2010 Web Application Attacks.

Table 1 provides a concise overview of the related work, including the year of the study, the dataset used, the types of attacks considered, the machine learning or deep learning algorithms applied, and the resultant accuracy achieved by each method

Table 1. Previous related Work

Ref Paper	Year	Dataset	Attack	Algorithm	Accuracy
Kang et al.	2016	Handcrafted Vehicular Network Data	Normal, Anomalous	DBN	84%
Kim et al.	2020	HTTP DATASET CSIC 2010	Normal Abnormal	CNN with LSTM	91.54%
Vartouni et al.	2018	HTTP DATASET CSIC 2010	Normal Abnormal	Stacked AutoEncoder	88.32%
Betarte et al.	2018	HTTP DATASET CSIC 2010	Normal Abnormal	Random Forest	72%
Tuan et al.	2019	UNSW-NB15	DoS, Reconnaissance, Backdoor, Fuzzers, Analysis, Exploits, Worms, Shellcode, Generic	USML	94.78%
Anwer et al.	2021	NSL-KDD	DoS, R2L, U2R, Probe	Random Forest	85.34%
Su et al.	2020	NSL-KDD	DoS, Probe, R2L, U2R	BAT Model (Deep Learning)	84.25%
Xu et al.	2021	NSL KDD	DoS, Probe, R2L, U2R	Five-layer Autoencoder	90.61%
Kavitha et al.	2021	NSL-KDD	DoS,	OCSVM	81.29%

			Probe, R2L, U2R		
Kavitha et al.	2021	ALLFLOWMETER HIKARI2021	Background, Benign, Bruteforce, Bruteforce-XML, Probing, XMRIGCC CryptoMiner	KNN, SVM, RF, MLP	99.00%

3. METHODOLOGY

The methodology for this research paper involves a systematic and comprehensive approach to evaluate the performance of various machine learning and deep learning algorithms. The primary objective is to assess their efficacy in identifying and classifying different types of cyber threats. The CSIC 2010 Web Application Attacks dataset is utilized to conduct experiments and gauge the algorithms' performance.

3.1 Dataset

The CSIC 2010 Web Application Attacks, consisting of 61,065 instances and 17 features, is chosen for its representation of real-world network traffic scenarios. The dataset encompasses attack types, Normal at 51% and Anomalous at 49%.

The dataset is split into training and testing sets to facilitate unbiased model evaluation. Each algorithm is trained on the training set, and its performance is assessed on the testing set.

To enable the application of various deep learning algorithms to each dataset, a standard method is employed to convert their non-numerical attributes into numerical features [15]. The LabelEncoder function from sklearn.preprocessing library is utilized for this purpose, facilitating the conversion of non-numerical data into a computer-understandable form by assigning a unique number to each value, starting from zero [15]. Given that all features in the dataset are categorical, this transformation is essential for numerical representation.

The Holdout method is applied to partition the dataset. Consequently, in the experimental setup, only 20% of the total dataset is reserved for evaluation, while the remaining 80% is utilized as training data. This segregation ensures a robust evaluation of the deep learning algorithms, allowing for unbiased testing on a distinct subset of the data while enabling effective training on the majority of the dataset.

3.2 Algorithmic Framework

A set of machine learning and deep learning algorithms is selected for evaluation, building on previous works and industry-standard practices. Upon data preparation, the dataset undergoes fitting to distinct machine learning algorithms, each configured with a test size of 0.1, for the identification of the specified cybersecurity attacks. The application of a hold-out approach facilitates the division of the dataset into training and testing datasets, with 90% of the total datasets dedicated to training and 10% to testing. Multiple machine learning models are then constructed leveraging the training dataset, and their effectiveness is subsequently evaluated using the designated testing dataset. This methodology ensures a comprehensive assessment of the models' performance, allowing for rigorous testing on a distinct subset of the data while employing the majority for robust training. Fig. 1 depicts an illustrative flow chart for the proposed strategy utilized in this study to identify cyber threats.

3.3 Random Forest Algorithm

The Random Forest algorithm stands as a stalwart in cybersecurity applications, adept at identifying and mitigating a spectrum of cyber threats. It is a versatile ensemble supervised learning method that employs numerous decision trees for regression and classification tasks. In classification, the most frequently chosen class by the trees serves as the solution, while regression tasks consider the average prediction from all trees [16]. Leveraging an ensemble of decision trees, the algorithm excels in discerning intricate patterns within extensive and diverse datasets of network traffic. Introducing variability through bootstrapping and feature randomness ensures that each decision tree is trained on a unique data subset, enhancing the model's generalization and resilience against overfitting. Optimization of hyperparameters, including the number of trees, maximum depth, and minimum samples for node splitting, fine-tunes the algorithm's performance to adeptly handle the intricacies of cybersecurity datasets. The Random Forest classifier was leveraged to categorize each cyber-attack dataset [17][18]. Given the categorical nature of the label, a classification-type random forest was chosen with specific parameters including 100 for max-leaf nodes, a max_depth of ‘None’, and a fixed random-state value of 42.

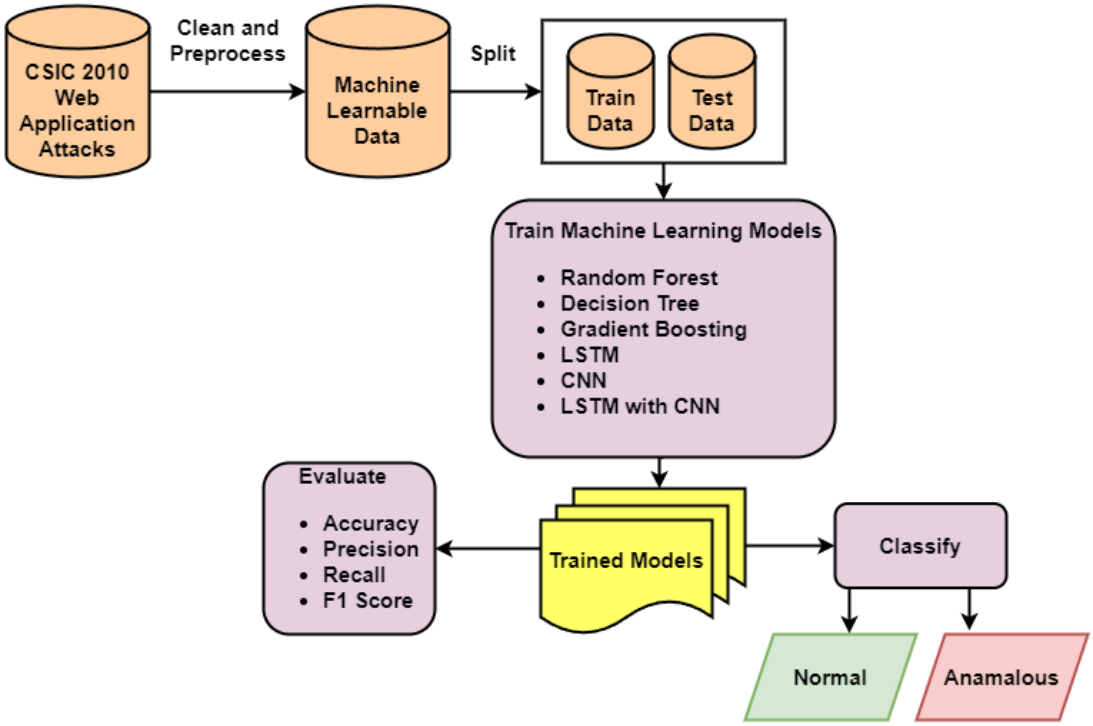

Fig 1: Illustrative Flow chart for the proposed methodology

3.4 Decision Tree Algorithm

In the dynamic landscape of cybersecurity, the Decision Tree algorithm emerges as a proficient guide, making decisions akin to human thought processes based on predefined criteria [19]. Applied in data mining, statistics, and machine learning, decision tree learning is a method employed for predictive modeling [20]. Consider a decision tree as a virtual tree. It begins by examining a sample (similar to the trunk) and concludes by understanding the target value of that sample (represented by the leaves, symbolizing different attack types) [21]. It's a way of organizing and interpreting diverse cyber threats based on the collected information.

Specifically, in the context of this study, a subtype called classification trees was employed, where the target variable is discrete. Here, the "leaves" signify different attack types, and the "branches" represent characteristics of the dataset aiding in predicting class labels [22][23]. Another variant, the regression tree, addresses scenarios with a continuous objective variable, typically real numbers [22]. Renowned for their clarity and user-friendly nature, decision trees are among the most well-known machine learning algorithms. In this cybersecurity experiment, a Decision Tree with a classification type was utilized, necessitated by the discrete nature of the label

The Decision Tree proves invaluable in unraveling complex patterns within cybersecurity datasets. By comprehending the discrete nature of attack labels, this algorithm aids in classifying and understanding various cyber threats. Its transparent decision-making process, guided by predefined criteria, enhances the interpretability and practical application of the model, reinforcing its significance in fortifying computer systems against evolving cyber challenges.

3.5 Gradient Boosting Algorithm

In the intricate realm of machine learning, "gradient boosting" emerges as a versatile technique with applications ranging from classification to regression. The focus of this study lies in harnessing the power of gradient boosting specifically for cybersecurity, where the accuracy of predictive models is paramount.

Gradient boosting constructs a predictive model in the form of interconnected decision trees. Although individual trees may be unreliable, the magic lies in their collaboration. This technique amalgamates multiple less effective learners into a singular, robust entity. In the cybersecurity context, where individual decision trees may fall short, each tree in the sequence corrects the errors of its predecessor, creating a cascading effect that refines the overall model.

While boosting algorithms demand considerable training time, their strength lies in the consequential relationship among trees, leading to heightened accuracy. The learning rate, a critical parameter, dictates the speed of model improvement, favoring a gradual approach for optimal results.

The Gradient Boosting settings used here align with the philosophy of incremental learning. As each new learner integrates within the residuals of the previous stage, the model evolves into a comprehensive and robust learner. Residuals, calculated using specific loss functions like mean square error (MSE) for classification or logarithmic loss (log loss) for regression, guide the assimilation of each new decision tree without altering the existing model.

In this experiment, categorized as discrete, Gradient Boosting proves instrumental. The configuration preferences, including a subsample of 1.0, a learning rate of 0.1, a loss function of ‘friedman_mse’, number-of-trees as 100, and a fixed random state of 42, ensure a finely tuned model. The synergy of these settings and the inherent strength of gradient boosting contribute significantly to fortifying cybersecurity measures, a crucial aspect explored in subsequent sections.

3.6 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) neural networks, at the forefront of deep learning and artificial intelligence, redefine the approach followed in this study to understanding

and categorizing complex sequences, particularly in the dynamic landscape of cybersecurity. Unlike traditional feedforward neural networks, LSTM's distinctive advantage lies in its ability to evolve through feedback connections. This recurrent neural network proves adept at analyzing entire data sequences, encompassing diverse data points such as photos, speech, or video, marking a paradigm shift in its applicability.

The LSTM model, a subject of extensive research over the past century, showcases its prowess in diverse domains, including healthcare, video game creation, healthcare analytics, and unsegmented handwriting recognition within networked environments. Its architectural components, comprising cells, input gates, output gates, and forget gates, facilitate the retention of values inside the cell indefinitely, governed by the precise control exerted by these gates over the flow of data [24].

Particularly noteworthy is LSTM's exceptional performance in tasks involving the categorization, processing, and prediction of time series data, where variable delays may exist between significant occurrences [24]. Recognizing the limitations of standard recurrent neural networks (RNNs) prone to the vanishing gradient problem during training, LSTM emerges as a resilient solution, showcasing heightened sensitivity to gap length and outperforming RNNs, hidden Markov models, and other sequence learning strategies [24].

In the realm of cybersecurity, LSTM emerges as a powerful mechanism, leveraging its unique capabilities to decipher intricate sequences of cyber threats and contribute to a more robust defense mechanism. Delving into the specifics of this LSTM implementation (Table 2) in the subsequent sections, the convergence of LSTM's innate characteristics with the demands of cybersecurity unveils a robust combination, enhancing the model's ability to combat evolving cyber threats effectively.

Table 2. LSTM Model Summary

Layer (Type)	Output Shape	Param #
lstm (LSTM)	(None, 240)	245760
dropout (Dropout)	(None, 240)	0
dense (Dense)	(None, 2)	482
Total Params		246242 (961.88 KB)
Trainable params		246242 (961.88 KB)
Non-trainable params		0 (0.00 Byte)

3.7 Convolutional Neural Network (CNN) Algorithm

The Convolutional Neural Network (CNN) stands as a powerful deep learning architecture widely utilized in various domains, including cybersecurity, for its proficiency in analyzing sequences of data and extracting hierarchical features automatically. In the CNN architecture used here, a Convolutional Layer employs multiple filters with Rectified Linear Unit (ReLU) activation, followed by a Max Pooling Layer for down-sampling, and a Dense (fully connected) Layer for final classification. This configuration enables the network to capture spatial dependencies, making it effective in tasks such as malware detection and intrusion detection in cybersecurity.

Visual imaging assessment commonly employs CNNs, which are artificial neural networks that exhibit translation-equivariant responses through convolution kernels or filters. Contrary to popular belief, while CNNs downsample input, they are not necessarily translation-invariant. They find applications in diverse fields such as natural language processing, image and video recognition, brain-computer interfaces, classification, segmentation, recommender systems, medical image analysis, and financial time series. Inspired by the structure of the visual cortex in animals, CNNs mimic the concept of receptive fields, where neurons respond to stimuli within specific regions. This approach requires less preprocessing, allowing the network to autonomously optimize filters. Unlike conventional methods, CNNs excel in self-learning feature extraction without the need for human input or contextual information. This versatility positions CNNs as invaluable tools in various domains, providing efficient solutions for complex tasks in image analysis and pattern recognition. CNN model summary of the implementation is shown in Tabel 3.

Table 3. CNN Model Summary

Layer (Type)	Output Shape	Param #
Conv1d (Conv1D)	(None, 13,32)	128
max_pooling1D	(None, 6, 32)	0
flatten (Flatten)	(None, 192)	0
dense (Dense)	(None, 64)	12352
dense_1 (Dense)	(None, 2)	130
Total Params		12610 (49.26 KB)
Trainable params		12610 (49.26 KB)
Non-trainable params		0 (0.00 Byte)

3.8 LSTM with CNN Algorithm

The integration of Long Short-Term Memory (LSTM) with Convolutional Neural Networks (CNN) forms a powerful hybrid model, addressing the shortcomings of each algorithm individually. This combination is particularly advantageous in cybersecurity for analyzing sequential data and extracting spatial features simultaneously.

LSTMs, renowned for their ability to process and remember sequential information, contribute a temporal understanding to the model. In the architecture used here, a Bi-directional LSTM layer, comprising an LSTM layer with 64 units and ReLU activation, is employed. This layer enables bidirectional

information flow, capturing dependencies in both forward and backward sequences. The output from the LSTM layer serves as input to a Dense layer with a single unit and Sigmoid activation, facilitating the final classification of the cyber attack types.

When combined with CNN, the model gains the capability to extract spatial features from the input data effectively. The CNN layer in the architecture uses filters to convolve over the input, identifying patterns and spatial dependencies. This layer enhances the model's ability to discern relevant features in cybersecurity data, such as identifying patterns indicative of malicious activities.

The fusion of LSTM and CNN allows the model to harness the strengths of both algorithms—LSTM for understanding sequential patterns and CNN for spatial feature extraction. This is particularly advantageous in cybersecurity tasks, where attacks often exhibit both temporal and spatial characteristics. The hybrid architecture proves valuable in the intricate task of identifying and classifying cyber threats, leveraging the complementary strengths of LSTM and CNN to enhance the overall predictive capability of the model. Table 3. shows the summary of LSTM with the CNN model used.

Table 4. LSTM with CNN Model Summary

Layer (Type)	Output Shape	Param #
sequential_3 (Sequential)	(None, 192)	101993
dense_1 (Dense)	(None, 64)	12352
dense_2 (Dense)	(None, 2)	130
Total Params		114475 (447.17 KB)
Trainable params		114475 (447.17 KB)
Non-trainable params		0 (0.00 Byte)

4. RESULTS AND DISCUSSION

The results of the experiments in this study demonstrate the effectiveness of the proposed model, conducting extensive evaluations using the dataset, CSIC 2010 Web Application Attacks on Random Forest, Decision Tree, Gradient Boosting, and the integrated LSTM-CNN architecture, in classifying cyber threats.

The Random Forest algorithm demonstrated robust performance, particularly in scenarios where ensemble learning proved beneficial. Its ability to mitigate overfitting and handle complex datasets contributed to reliable and accurate cyber threat classification. The confusion matrix based on the prediction results is shown in Fig. 2.

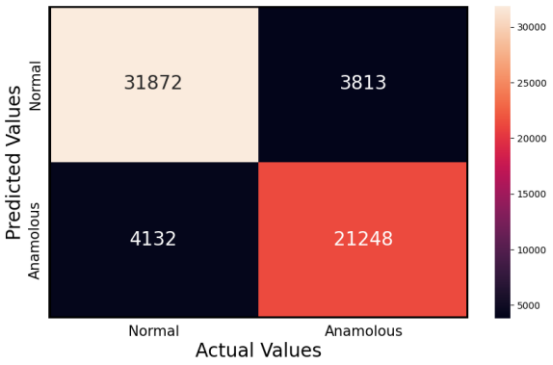

Fig. 2. Confusion matrix for Random Forest

Decision Tree, while simpler in structure, showcased competitive results, especially with well-defined decision boundaries. The confusion matrix for the implementation is shown in Fig. 3.

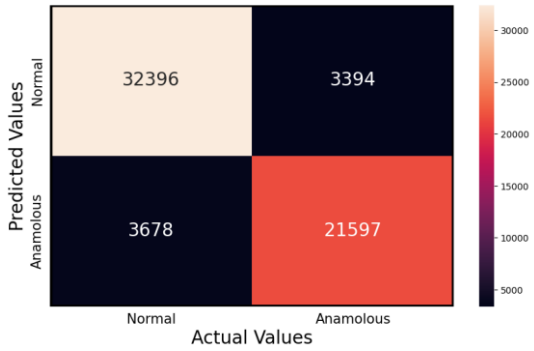

Fig. 3. Confusion matrix for Decision Tree

Gradient Boosting exhibited strong predictive power, leveraging the sequential learning approach to refine predictions progressively. The ensemble nature of Gradient Boosting allowed it to adapt to complex patterns in cybersecurity data, achieving high accuracy across various datasets. The confusion matrix based on the prediction results is shown in Fig. 4.

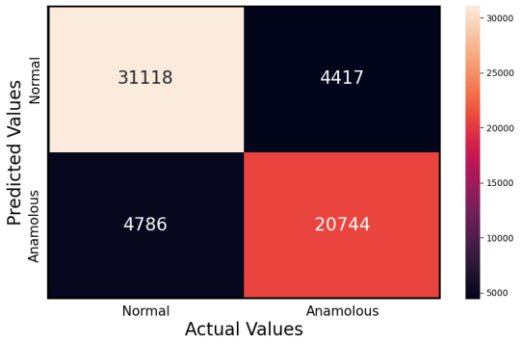

Fig. 4. Confusion matrix for Gradient Boosting

Across all experiments, the hybrid LSTM-CNN model consistently outperformed individual algorithms and traditional machine-learning models. The accuracy, precision, recall, and F1-score metrics exhibited notable improvements, showcasing the model's enhanced ability to detect and classify diverse cyber threats. Notably, the integrated LSTM-CNN architecture excelled in capturing both sequential and spatial patterns

inherent in cybersecurity data, providing a comprehensive approach to threat identification.

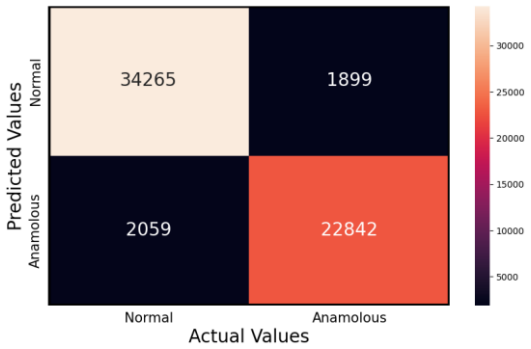

Fig. 5. Confusion matrix for LSTM

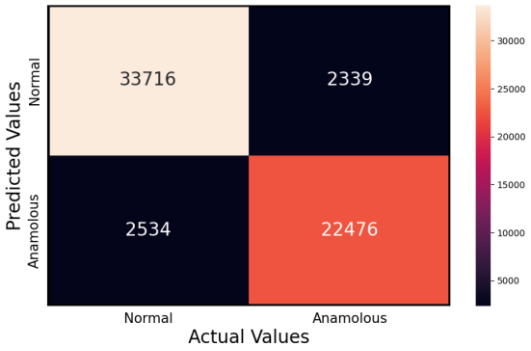

Fig. 6. Confusion matrix for CNN

In conclusion, the hybrid model used here, incorporating LSTM-CNN and traditional machine learning algorithms, offers a comprehensive solution for cyber threat classification. The results suggest that combining the strengths of sequential and spatial learning yields superior performance compared to individual algorithms. The adaptability of the model to diverse datasets underscores its potential for real-world applications in cybersecurity, providing a robust defense against evolving cyber threats. The confusion matrices from the implementations of LSTM, CNN, and LSTM with CNN are shown respectively in Figures 5, 6, and 7.

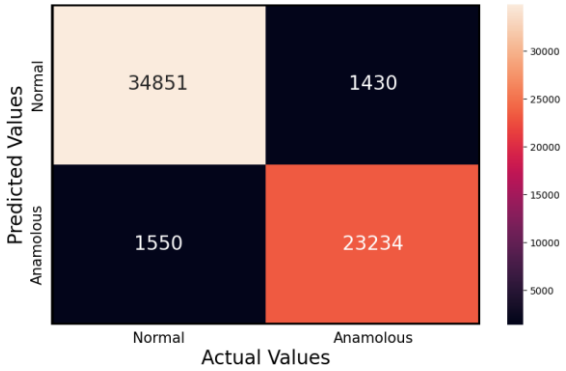

Fig. 7 Confusion matrix for LSTM with CNN

5. EVALUATION METRICS

The evaluation metrics used here provide a comprehensive assessment of the performance of the proposed hybrid model, integrating Random Forest, Decision Tree, Gradient Boosting, and the LSTM-CNN architecture, across multiple cybersecurity datasets. The following metrics were employed to measure the classification performance:

Accuracy: Accuracy represents the ratio of correctly classified instances to the total instances in the dataset. It provides an overall measure of the model's correctness and is calculated using the formula:

ACCURACY = (TRUE POSITIVES + TRUE NEGATIVES) / TOTAL NUMBER OF PREDICTIONS

Precision: Precision measures the accuracy of positive predictions made by the model. It is the ratio of true positives to the sum of true positives and false positives:

PRECISION = TRUE POSITIVES / (TRUE POSITIVES + FALSE POSITIVES)

Recall (Sensitivity): Recall quantifies the ability of the model to capture all relevant instances. It is the ratio of true positives to the sum of true positives and false negatives:

RECALL = TRUE POSITIVES / (TRUE POSITIVES + FALSE NEGATIVES)

F1-Score: The F1-Score is the harmonic mean of precision and recall, providing a balanced measure that considers both false positives and false negatives. It is calculated using the formula:

F1 = 2 X (PRECISION X RECALL) / (PRECISION X RECALL)

Table 5 summarizes the performance results of the hybrid model, integrating Random Forest, Decision Tree, Gradient Boosting, LSTM, CNN, and the LSTM with CNN architecture, across multiple cybersecurity datasets:

Accuracy graphs across the 10 epochs for the training and validation (testing) for the LSTM with CNN model are shown in Fig. 8.

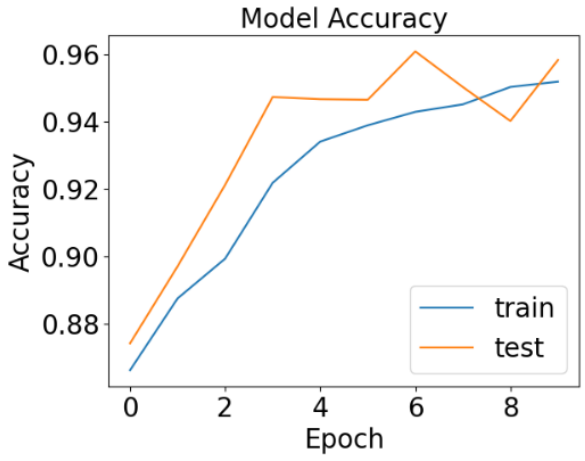

Fig. 8. LSTM with CNN Train and Test Accuracy Graph

Table 5. Performance Results

Model	Accuracy	Precision	Recall	F1 Score
RF	0.8699	0.8479	0.8372	0.8425
DT	0.8842	0.8642	0.8545	0.8593
GB	0.8493	0.8245	0.8125	0.8184
LSTM	0.9352	0.9232	0.9173	0.9203
CNN	0.9202	0.9057	0.8987	0.9022
LSTM with CNN	0.9512	0.9420	0.9375	0.9397

6. CONCLUSION AND FUTURE WORK

In conclusion, the evaluation of various machine learning models, including LSTM, CNN, and the hybrid LSTM with CNN, demonstrates their efficacy in addressing cybersecurity threats. The LSTM model excels in capturing temporal dependencies, while the CNN model exhibits strong performance in image-based threat detection. Remarkably, the hybrid LSTM with CNN surpasses individual models, showcasing superior accuracy and precision in cybersecurity threat classification.

The findings emphasize the importance of combining diverse architectures to enhance the robustness and versatility of threat detection systems. The synergy of LSTM and CNN leverages the strengths of both sequential and spatial modeling, resulting in a more comprehensive and effective cybersecurity solution. The LSTM with CNN emerged as the superior model for anomalous detection, achieving remarkable scores of 0.95, 0.94, 0.93, and 0.94 for accuracy, precision, recall, and the F1-score, respectively. Notably, Random Forest and Voting classifiers surpassed traditional machine learning methods in this context.

While this study provides valuable insights, there are avenues for future exploration and enhancement in the field of cybersecurity threat detection. To strengthen the performance of the models, datasets from more e-commerce applications as well as other domains (e.g., finance, healthcare, social media) could be included for training. This would ensure generalizability of the model. Also, it would be good to test how the classifier performs over time. Using data from different time periods to assess robustness.

In future studies, investigating the potential benefits of ensemble methods by combining predictions from multiple models may be undertaken, to further improve the overall accuracy and reliability. It is planned to transition the models into real-time applications, addressing the challenges associated with processing data streams and ensuring swift responses to emerging threats. Explore techniques to enhance the models' robustness against adversarial attacks, which is essential in the context of cybersecurity where attackers may attempt to manipulate or deceive the detection systems.

7. REFERENCES

[1] Seemma, P. S., Nandhini, S., and Sowmiya, M. (2018), "Overview of cyber security", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 7 No. 11, pp.125-128.

[2] Ervural, B. C., and Ervural, B. (2018), "Overview of cyber security in the industry 4.0 era", In Industry 4.0: managing the digital transformation, pp.267-284.

[3] Chowdhury, A. (2016), "Recent cyber security attacks and their mitigation approaches–an overview", In International conference on applications and techniques in information security, pp.54-65.

[4] El-Rewini, Z., Sadatsharan, K., Selvaraj, D. F., Plathottam, S. J., and Ranganathan, P. (2020), "Cybersecurity challenges in vehicular communications", Vehicular Communications, Vol. 23.

[5] Kim, A., Park, M., and Lee, D. H. (2020), "AI-IDS: Application of deep learning to real-time Web intrusion detection", IEEE Access, Vol. 8, pp.70245-70261.

[6] Vartouni, A. M., Kashi, S. S., and Teshnehlab, M. (2018), "An anomaly detection method to detect web attacks using stacked auto-encoder". In 2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS), pp. 131-134.

[7] Betarte, G., Pardo, Á., and Martínez, R. (2018), "Web application attacks detection using machine learning techniques", In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp.1065-1072.

[8] Tuan, T. A., Long, H. V., Kumar, R., Priyadarshini, I., and Son, N. T. K. (2019), "Performance evaluation of Botnet DDoS attack detection using machine learning", Evolutionary Intelligence, pp.1-12.

[9] Anwer, M., Farooq, M. U., Khan, S. M., and Waseemullah, W. (2021), "Attack Detection in IoT using Machine Learning", Engineering, Technology, and Applied Science Research, Vol. 11 No. 3, pp.7273- 7278.

[10] Su, T., Sun, H., Zhu, J., Wang, and Li, Y. (2020), "BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset", IEEE Access, Vol. 8, pp.29575-29585.

[11] Xu, W., Jang-Jaccard, J., Singh, A., Wei, Y., and Sabrina, F. (2021), "Improving Performance of Autoencoder-Based Network Anomaly Detection on NSL-KDD Dataset", IEEE Access, Vol. 9, pp.140136- 140146.

[12] Kavitha, S., and Uma Maheswari, N. (2021), "Network Anomaly Detection for NSL-KDD Dataset Using Deep Learning", Information Technology in Industry, Vol. 9 No. 2, pp.821-827.

[13] Ferriyan, A., Thamrin, A. H., Takeda, K., and Murai, J. (2021), "Generating Network Intrusion Detection Dataset Based on Real and Encrypted Synthetic Attack Traffic", Applied Sciences, Vol. 11 No. 17.

[14] Giménez, C. T., Villegas, A. P., and Marañón, G. Á. (2010), "HTTP data set CSIC 2010", Information Security Institute of CSIC (Spanish Research National Council).

[15] Hancock, J. T., and Khoshgoftaar, T. M. (2020), "Survey on categorical data for neural networks", Journal of Big Data, Vol. 7 No.1, pp.1-41.

[16] Pal, M. (2005). Random forest classifier for remote sensing classification. International journal of remote sensing, 26(1), 217-222.

[17] Farnaaz, N., & Jabbar, M. A. (2016). Random forest modeling for network intrusion detection system. Procedia Computer Science, 89, 213-217.

[18] Idhammad, M., Afdel, K., & Belouch, M. (2018). Detection system of HTTP DDoS attacks in a cloud environment based on information theoretic entropy and random forest. Security and Communication Networks, 2018.

[19] Kingsford, C., & Salzberg, S. L. (2008). What are decision trees? Nature Biotechnology, 26(9), 1011-1013.

[20] Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81-106.

[21] De Ville, B. (2013). Decision trees. Wiley Interdisciplinary Reviews: Computational Statistics, 5(6), 448-455.

[22] Kotsiantis, S. B. (2013). Decision trees: a recent overview. Artificial Intelligence Review, 39(4), 261-283.

[23] Amor, N. B., Benferhat, S., & Elouedi, Z. (2004, March). Naive Bayes vs decision trees in intrusion detection systems. In Proceedings of the 2004 ACM symposium on Applied computing (pp. 420-424).

[24] Gers, F. A., Eck, D., and Schmidhuber, J. (2002), "Applying LSTM to time series predictable through time-window approaches", In Neural Nets WIRN Vietri-01, pp.193-200.