https://doi.org/10.33472/AFJBS.6.6.2024.6250-6264

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

A Systematic Framework for Cyberthreat Detection Using Machine Learning Algorithms

Dr.S.Gnanamurthy[1], V. Gangadhar[2]

[1]Associate Professor, Department of Computer Science and Engineering, Kuppam Engineering College, KES Nagar, Kuppam, Andhra Pradesh 51742, India.
[2]PG Scholoar, Department of Computer Science and Engineering, Kuppam Engineering College, KES Nagar, Kuppam, Andhra Pradesh 51742, India.

Email: [1]gnanamurthyspec@gmail.com, [2]vgangadharvgr@gmail.com

Article Info

Volume 6, Issue 6, June 2024

Received: 23 April 2024

Accepted: 31 May 2024

Published: 26 June 2024

doi: 10.33472/AFJBS.6.6.2024.6250-6264

ABSTRACT:

In this study, we present a systematic methodology for cyberthreat detection leveraging machine learning algorithms. The process begins with data preparation, including loading, pre-processing, normalization, and splitting into training and validation sets. Feature extraction is performed using a Variational Autoencoder (VAE), reducing the dimensionality of the data to 20 features. Subsequently, feature selection techniques such as Variance Threshold Filter, KBest with Chi2 Filter and KBest with Mutual Information Filter are applied to further refine the feature space to 15 features. For model selection and evaluation, various algorithms including Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, K-Nearest Neighbors, Extra Trees, Naive Bayes, and Linear SVC are evaluated. Through initial exploration, K-Nearest Neighbors, Extra Trees, Naive Bayes, and Linear SVC emerge as potential candidates. Hyper parameter tuning is conducted for Logistic Regression, Naive Bayes, and Linear SVC using Randomized Search CV. Further evaluation entails assessing the tuned models using multiple evaluation metrics such as accuracy, negative log loss, and ROC AUC score. ROC curves and confusion matrices are plotted to gain a comprehensive understanding of each model's performance. Based on the evaluation results, Logistic Regression and Naive Bayes are selected as the final models. Validation of the selected models is carried out on the validation set, and detailed classification reports are provided. Overall, our approach offers a structured framework for building and evaluating machine learning models for cyberthreat detection. The documentation provided throughout the process enhances transparency and facilitates comprehension of each step and decision rationale

Keywords: Cyber threat Detection, Machine Learning, Variational Auto encoder (VAE), Feature Selection, Logistic Regression, Naive Bayes, Hyperparameter Tuning etc.

1. Introduction

In the modern digital landscape, cyber threats pose significant risks to individuals, organizations, and nations. The increasing complexity and frequency of these threats necessitate advanced methods for detection and prevention. Traditional approaches to cyber security, often reliant on rule-based systems and manual oversight, struggle to keep pace with the evolving tactics of cybercriminals. As a result, there is a growing interest in leveraging machine learning (ML) to enhance the effectiveness and efficiency of cyberthreat detection. Machine learning algorithms can analyze vast amounts of data to identify patterns and anomalies indicative of cyberthreats. These algorithms, when properly trained and tuned, offer the potential for real-time detection and adaptive defense mechanisms. This study aims to develop a robust methodology for cyberthreat detection using a systematic approach to machine learning.

The process begins with meticulous data preparation, including normalization and the division of data into training and validation sets. Feature extraction is performed using Variational Autoencoders (VAE) to reduce dimensionality and enhance the representation of the data. Following this, feature selection techniques such as the Variance Threshold Filter, KBest with Chi2 Filter, and KBest with Mutual Information Filter are employed to refine the feature space further.

Several machine learning algorithms, including Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, K-Nearest Neighbors, Extra Trees, Naive Bayes, and Linear Support Vector Classifier (SVC), are evaluated to identify the most promising models for cyberthreat detection. Initial explorations highlight K-Nearest Neighbors, Extra Trees, Naive Bayes, and Linear SVC as potential candidates. Hyperparameter tuning, conducted via Randomized Search CV, optimizes the performance of Logistic Regression, Naive Bayes, and Linear SVC models.

The evaluation phase assesses the tuned models using multiple metrics such as accuracy, negative log loss, and ROC AUC score. Detailed analyses, including ROC curves and confusion matrices, provide a comprehensive understanding of model performance. Based on these evaluations, Logistic Regression and Naive Bayes are selected as the final models.
Validation of these models on a separate validation set ensures their effectiveness and reliability in real-world scenarios. This structured methodology demonstrates the potential of machine learning in enhancing cyberthreat detection and provides a transparent, reproducible framework for future research and implementation. The comprehensive documentation of each step and the rationale behind key decisions further contribute to the field's knowledge base, offering valuable insights for practitioners and researchers alike.
The organizational framework of this study divides the research work in the different sections. The Literature survey is presented in section 2. In section 3 discussed about proposed system methodologies. Further, in section 4 shown Results is discussed and. Conclusion and future work are presented by last sections 5.

2. Literature Survey

The field of cyberthreat intelligence has seen significant advancements, driven by the growing need to protect digital infrastructure from sophisticated cyberattacks. This literature survey delves into various aspects of cyberthreat intelligence, examining its evolution,

methodologies, challenges, and the role of machine learning in enhancing threat detection and response.

The SolarWinds cyberattack, a significant incident linked to Russian espionage tools, exemplifies the complexity and severity of modern cyber threats. This breach, which infiltrated numerous government and private sector systems, underscores the necessity for advanced detection mechanisms and robust cybersecurity strategies (Source: [1]).

Threat intelligence involves the collection, analysis, and dissemination of information regarding potential cyber threats. R. McMillan (Source: [2]) provides a comprehensive definition, highlighting the role of threat intelligence in enabling organizations to anticipate, identify, and mitigate cyber risks proactively.

D. Shackleford (Source: [3]) explores how organizations utilize cyberthreat intelligence to enhance their cybersecurity posture. The report from the SANS Institute highlights the practical applications of threat intelligence and the benefits of integrating it into security operations.

H. Dalziel (Source: [4]) discusses the critical components and best practices for developing an effective cyber threat intelligence capability. This work provides a detailed guide on establishing robust threat intelligence frameworks, emphasizing the importance of strategic planning and resource allocation.

C. Fachkha and M. Debbabi (Source: [5]) provide a thorough survey on utilizing the Darknet as a source of cyber intelligence. Their study categorizes and characterizes the types of intelligence that can be gathered from these hidden parts of the internet, which are often rich in information about emerging threats and attacker behaviors.

The work by J. Robertson et al. (Source: [6]) extends the understanding of the Darkweb as a source of valuable cyber intelligence. Their research focuses on methods for mining and analyzing data from the Darkweb to uncover actionable threat insights.

W. Tounsi and H. Rais (Source: [7]) survey the landscape of technical threat intelligence, emphasizing the significance of sharing threat data among organizations to counter sophisticated cyber attacks. Their study highlights various techniques for collecting and disseminating technical threat information.

T. D. Wagner et al. (Source: [8]) discuss the challenges and benefits of sharing cyber threat intelligence. Their research points to the need for collaborative efforts and standardized protocols to enhance the effectiveness of shared threat information.

M. S. Abu et al. (Source: [9]) and A. Ibrahim et al. (Source: [10]) identify several challenges in leveraging threat intelligence, such as data quality issues, integration difficulties, and the dynamic nature of cyber threats. These studies stress the importance of overcoming these obstacles to maximize the utility of threat intelligence.

M. R. Rahman et al. (Source: [11], [12]) explore the automation of threat intelligence extraction from unstructured texts. Their work involves leveraging natural language processing (NLP) and machine learning to streamline the process of converting raw data into actionable intelligence, thus keeping pace with the rapidly changing threat landscape.

R. Brown and P. Stirparo (Source: [13]) provide comprehensive surveys on the current practices and future directions in cyber threat intelligence. These surveys, conducted by the SANS Institute, offer valuable insights into the adoption, benefits, and limitations of threat intelligence practices across various organizations.

Dataset

The The Aegean WiFi Intrusion/Threat Dataset (AWID2) serves as a comprehensive resource for evaluating intrusion detection systems and machine learning algorithms within the context of WiFi network security. Collected from WiFi networks within controlled environments, AWID2 captures a diverse array of network traffic encompassing various WiFi-related

intrusions and threats. These may include rogue access point detection, deauthentication attacks, authentication bypass attempts, and other malicious activities targeting WiFi networks. The dataset comprises features such as source and destination MAC addresses, IP addresses, WiFi signal strength (RSSI), packet types, and payload contents from WiFi frames. Prior to analysis, preprocessing steps are necessary, involving data cleaning, feature engineering, normalization, and handling of class imbalance. By splitting the dataset into training, validation, and testing sets, researchers and practitioners can train and evaluate machine learning models effectively to detect and mitigate WiFi network intrusions and threats, ultimately contributing to the enhancement of WiFi network security measures.

Data Visualization

	frame.interface_id	frame.dlt	frame.offset_shift	frame.time_delta	\
0	0	0	0	6.570e-05	
1	0	0	0	1.430e-05	
2	0	0	0	3.553e-02	
3	0	0	0	5.128e-03	
4	0	0	0	3.512e-02	

	frame.time_delta_displayed	frame.len	frame.cap_len	frame.marked	\
0	6.570e-05	0.009	0.009	0	
1	1.430e-05	0.000	0.000	0	
2	3.553e-02	0.071	0.071	0	
3	5.128e-03	0.095	0.095	0	
4	3.512e-02	0.071	0.071	0	

Correlation

wlan.fc.subtype	-0.839
radiotap.channel.type.ofdm	-0.652
radiotap.datarate	-0.619
wlan.seq	-0.494
wlan.qos.priority	-0.453
wlan.qos.tid	-0.453
frame.len	-0.438
frame.cap_len	-0.438
data.len	-0.435
wlan.duration	-0.296
wlan.fc.retry	-0.285
wlan_mgt.fixed.reason_code	-0.257
wlan.da	-0.202
wlan_mgt.tim.dtim_period	-0.183
wlan_mgt.fixed.capabilities.preamble	-0.168
wlan_mgt.fixed.timestamp	-0.151
wlan_mgt.rsn.akms.count	-0.148
wlan_mgt.rsn.version	-0.148
wlan_mgt.rsn.akms.type	-0.148
wlan_mgt.rsn.gcs.type	-0.148
wlan_mgt.rsn.pcs.count	-0.147
wlan.wep.key	-0.104
wlan.ra	-0.100
wlan.tkip.extiv	-0.092
wlan_mgt.tim.dtim_count	-0.068
wlan_mgt.fixed.listen_ival	-0.059
wlan_mgt.fixed.aid	-0.056
wlan_mgt.fixed.capabilities.privacy	-0.054
wlan_mgt.fixed.auth_seq	-0.054
wlan_mgt.tagged.all	-0.042
wlan_mgt.country_info.environment	-0.039

Skew

```
wlan.fcs_good                          -67.958
radiotap.antenna                       -67.958
radiotap.flags.fcs                     -67.958
radiotap.present.rxflags               -67.958
radiotap.channel.type.2ghz             -67.958
radiotap.present.dbm_antsignal         -67.958
radiotap.present.channel               -67.958
radiotap.present.antenna               -67.958
radiotap.present.tsft                  -67.958
radiotap.length                        -67.958
dtype: float64
wlan.ba.control.cbitmap                          110.127
wlan.ba.control.ackpolicy                        127.167
wlan_mgt.fixed.sequence                          148.691
wlan_mgt.fixed.category_code                     179.850
wlan.ba.bm                                       181.116
wlan_mgt.fixed.current_ap                        211.958
wlan.bar.type                                    220.274
wlan_mgt.tcprep.trsmt_pow                        311.519
wlan_mgt.rsn.capabilities.ptksa_replay_counter   311.519
wlan_mgt.fixed.capabilities.spec_man             311.519
dtype: float64
```

The volume of features creates difficulty in meaningful visualisation. Two step approach next:

1. Use SelectKBest and chi2 to select top 10 attributes for review.
2. Split the data into the four categories based on the prefix of the feature names: -frame -radiotap -wlan -wlan_mgmt

Selecting 10 Attributes

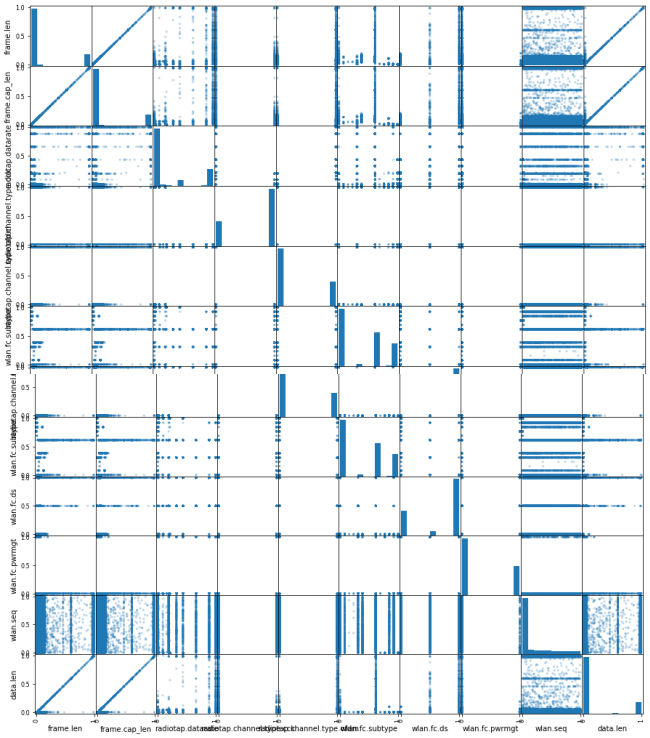


Fig. 1. Showing atrributes

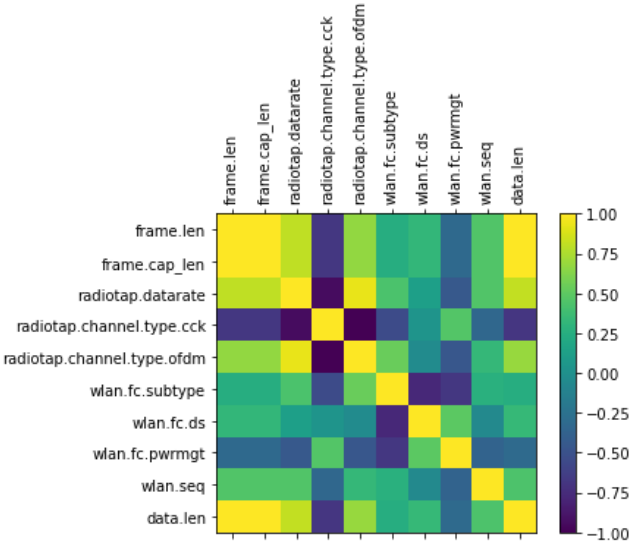


Fig. 2. Showing Correlation

This figure 1 illustrates the distribution of defects within the dataset. The number of occurrences for both true (defective) and false (non-defective) instances is displayed. The histogram depicts the frequency of defects, providing insight into the imbalance between defective and non-defective software artifacts.

3. Proposed Method

The proposed method for cyberthreat detection leverages the Aegean WiFi Intrusion/Threat Dataset (AWID2) and advanced machine learning techniques to develop an effective intrusion detection system. The process begins with extensive data preparation, including cleaning, normalization, and imputation, followed by splitting the dataset into training, validation, and testing sets. A Variational Autoencoder (VAE) is then employed to extract essential features, reducing the data to 20 significant features. This is followed by feature selection using Variance Threshold Filter, KBest with Chi-Squared Filter, and KBest with Mutual Information Filter, narrowing the dataset to the 15 most informative features. Various machine learning algorithms are evaluated, with K-Nearest Neighbors, Extra Trees, Naive Bayes, and Linear SVC identified as strong candidates. Hyperparameter tuning using RandomizedSearchCV further refines Logistic Regression, Naive Bayes, and Linear SVC models. The models are evaluated on metrics such as accuracy, negative log loss, and ROC AUC score, with ROC curves and confusion matrices providing detailed insights. Logistic Regression and Naive Bayes are selected as the final models, validated on the testing set, and shown to offer reliable performance for real-world cyberthreat detection in WiFi networks.

A. System Architecture

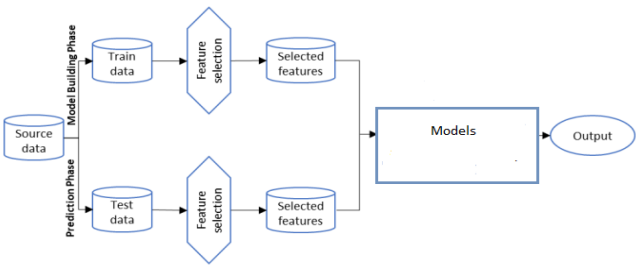


Fig. 3. System Architecture

The proposed method for cyberthreat detection based on the Aegean WiFi Intrusion/Threat Dataset (AWID2) involves several stages. Each stage is critical in developing a robust and effective intrusion detection system (IDS) using machine learning techniques. Below is a detailed explanation of each stage represented in the block diagram.

1) Data Collection and Preprocessing

Data Collection: WiFi network traffic data is collected from various sources within a controlled environment where different types of attacks are simulated.

- Data Cleaning: The raw data is cleaned by removing duplicates, irrelevant information, and outliers to ensure the dataset's quality.
- Normalization: Numerical features are scaled to a common range, typically between 0 and 1, to prevent features with larger magnitudes from dominating the model training process.
- Imputation: Missing values in the dataset are handled using imputation techniques to ensure no loss of important information.

2. Feature Extraction

- Variational Autoencoder (VAE): A VAE is used for feature extraction, reducing the dimensionality of the dataset to 20 significant features. This step helps in capturing the essential characteristics of the data while removing noise and redundancy.

3. Feature *Selection*

- Variance Threshold Filter: Features with low variance are removed as they are unlikely to contribute significantly to the model's predictive power.
- KBest with Chi-Squared (Chi2) Filter: This method selects features based on their statistical significance with respect to the target variable.
- KBest with Mutual Information Filter: This filter selects features that capture the most relevant information about the target variable.
- Final Feature Set: After applying these three techniques, the feature set is reduced to the 15 most informative features.

4. Model Selection and Evaluation

- Initial Model Training: Various machine learning algorithms are trained on the preprocessed and reduced dataset. These algorithms include Logistic Regression, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), K-Nearest Neighbors (KNN), Extra Trees, Naive Bayes, and Linear Support Vector Classifier (SVC).
- Model Evaluation: The performance of these models is evaluated using metrics such as accuracy, negative log loss, and ROC AUC score.
- Identify Best Candidates: Based on the initial evaluations, KNN, Extra Trees, Naive Bayes, and Linear SVC are identified as strong candidates for further tuning.

5. Hyperparameter Tuning

- *RandomizedSearchCV:* Hyperparameter tuning is performed using RandomizedSearchCV to optimize the parameters of Logistic Regression, Naive Bayes, and Linear SVC models, aiming to improve their performance.

6. Further Evaluation

- Performance Metrics: The tuned models are re-evaluated using accuracy, negative log loss, and ROC AUC score.
- Visualization: ROC curves and confusion matrices are plotted to provide a comprehensive understanding of each model's performance, highlighting their ability to distinguish between normal and malicious traffic.

7. Final Model Selection

- Select Final Models: Logistic Regression and Naive Bayes are selected as the final models based on their overall performance in the evaluations.

8. Validation

- Model Validation: The final models are validated on the testing set to ensure their effectiveness and reliability in real-world scenarios.
- Classification Reports: Detailed classification reports are generated to summarize the performance metrics, providing insights into precision, recall, F1-score, and overall accuracy.

4. Methodology

The proposed methodology for cyberthreat detection using the Aegean WiFi Intrusion/Threat Dataset (AWID2) involves a structured and systematic approach encompassing several key stages. Initially, the data undergoes extensive preparation, including cleaning to remove duplicates and irrelevant information, normalization to scale numerical features, and imputation to handle missing values, followed by splitting into training, validation, and testing sets. Feature extraction is then performed using a Variational Autoencoder (VAE) to reduce the dimensionality to 20 significant features. Subsequently, a rigorous feature selection process is applied using Variance Threshold Filter, KBest with Chi-Squared Filter, and KBest with Mutual Information Filter, reducing the feature set to the 15 most informative features. Various machine learning algorithms, including Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, K-Nearest Neighbors, Extra Trees, Naive Bayes, and Linear SVC, are evaluated based on metrics such as accuracy, negative log loss, and ROC AUC score. The best-performing models—K-Nearest Neighbors, Extra Trees, Naive Bayes, and Linear SVC—are identified for further tuning using RandomizedSearchCV. After hyperparameter tuning, the models are re-evaluated, and detailed performance metrics, including ROC curves and confusion matrices, are analyzed. Finally, Logistic Regression and Naive Bayes are selected as the final models, validated on the testing set, and detailed classification reports are generated. This comprehensive methodology ensures the development of a robust and effective intrusion detection system tailored to the unique challenges of WiFi network security

B. Implementation

1) Data Preparation:

- Load Dataset: Load AWID2 dataset and clean it by removing duplicates and handling missing values.
- Normalize Data: Use StandardScaler to normalize the features.
- Split Data: Split the data into training, validation, and testing sets.

2) Feature Extraction:

Variational Autoencoder (VAE): Implement a VAE to reduce dimensionality to 20 features.

3) Feature Selection:

- Variance Threshold: Apply Variance Threshold Filter to select relevant features.
- KBest Filters: Use KBest with Chi-Squared and Mutual Information filters to further reduce features to 15.

4) Model Training:

- Train Models: Train various models (Logistic Regression, LDA, QDA, KNN, Extra Trees, Naive Bayes, Linear SVC) on the training data.
- Evaluate Models: Evaluate models using accuracy, log loss, and ROC AUC score.

5) Hyperparameter Tuning:

- RandomizedSearchCV: Tune hyperparameters for Logistic Regression, Naive Bayes, and Linear SVC.

6) Model Evaluation:

- Evaluate Tuned Models: Use validation set to evaluate the tuned models and plot ROC curves and confusion matrices.

7) Final Model Selection:

- Select Best Models**:** Select Logistic Regression and Naive Bayes as the final models based on evaluation metrics.

8) Model Validation:

- Validate Models: Validate the selected models on the testing set and generate detailed classification reports.

C. Flow chart

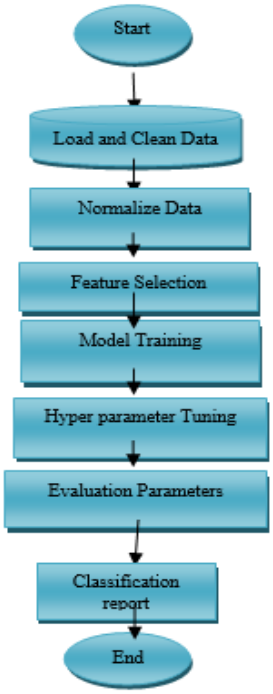


Fig. 4. Implmentation Flow Diagram

D. Performance Metrics

Performance measures are used to evaluate the network performance of the proposed model. This work uses accuracy, precision, recall and f1-score as performance measure, which are formulated.

a) Accuracy:

Measures the overall correctness of recognized signs or gestures compared to the ground truth.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ Number\ of\ Predictions} \quad (1)$$

b) Precision:

Precision signifies the proportion of correctly recognized signs among all recognized signs.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Where
TP=True Positives
FP= False Positives

c) Recall:

Recall measures the proportion of correctly recognized signs among all actual signs

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Where
TP=True Positives
FP= False Positives
FN=False Negatives

d) F1-Score:

Harmonic mean of precision and recall, providing a balanced measure of a model's performance

$$F1 - Score = 2\ X \frac{Precision\ X\ Recall}{Precision + \ Recall} \quad (4)$$

5. Results and Discussion

The comparison tables offer a detailed insight into the performance of various machines learning models dataset.

E. Logistic RegressionMachine Learning Algorithm

The simulation results are critical for understanding the effectiveness of Logistic Regression in identifying cyber threats. Below is an explanation of the key findings, as depicted in the AUC graph and the confusion matrix.

The AUC (Area Under the Curve) graph is a performance measurement tool for binary classification models. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. The AUC value quantifies the overall ability of the model

to discriminate between the positive class (cyberthreat) and the negative class (non-cyberthreat).
AUC Value is 0.997 is very close to 1, indicating that the Logistic Regression model has excellent discrimination capability. An AUC of 0.997 suggests that the model can correctly distinguish between cyber threats and non-threats with a very high degree of accuracy.

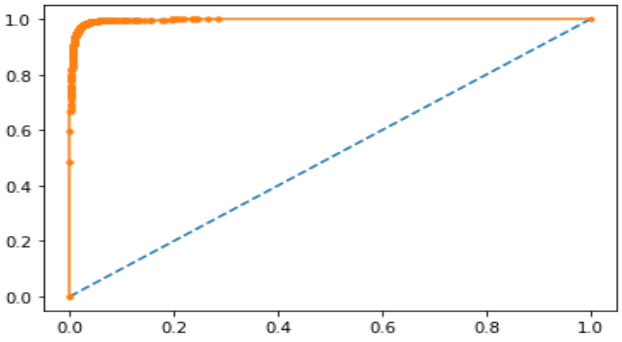


Fig. 5. AUC Graph for using logistic Regression

The confusion matrix provides a detailed breakdown of the model's performance in term s of the number of true and false predictions for both the positive and negative classes.
The high number of true positives (9488) and true negatives (9464) indicates that the model is highly effective at correctly identifying both cyberthreats and non-threats. The relatively low number of false positives (274) and false negatives (183) suggests that the model has a low error rate in its predictions, minimizing the risk of false alarms and missed threats. The model achieves an overall accuracy of approximately 97.64%, derived from the total number of correct predictions (TP + TN) divided by the total number of predictions. This high accuracy indicates that the model is reliable and robust in a real-world cyberthreat detection scenario.

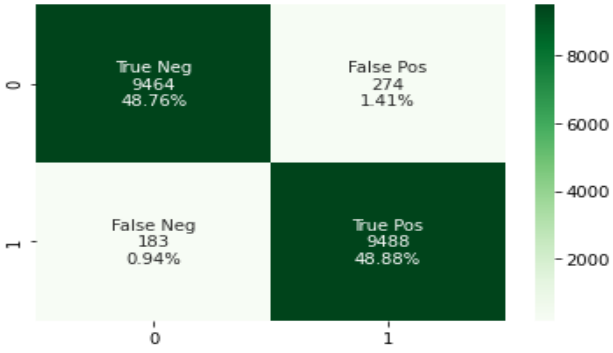


Fig. 6. Confusion Matrix

	precision	recall	f1-score	support
0.0	0.98	0.97	0.98	9738
1.0	0.97	0.98	0.98	9671
accuracy			0.98	19409
macro avg	0.98	0.98	0.98	19409
weighted avg	0.98	0.98	0.98	19409

Fig. 7. Hyper Tunning Parameters

F. Naïve bayes Machine Learning algorithm

AUC Value 0.997 is very close to 1, indicating that the Naive Bayes model has an excellent ability to distinguish between cyber threats and non-threats. An AUC of 0.997 suggests that the model can accurately differentiate between the two classes with a very high degree of precision

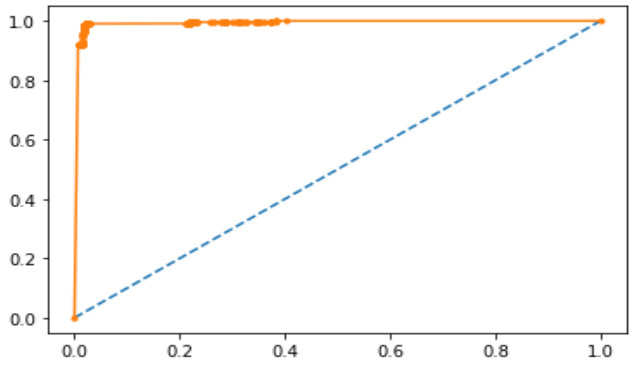


Fig. 8. AUC Graph

The high number of true positives (9439) and true negatives (9561) indicates that the Naive Bayes model is highly effective at correctly identifying both cyberthreats and non-threats. The relatively low number of false positives (232) and false negatives (177) suggests that the model has a low error rate in its predictions, minimizing the risk of false alarms and missed threats.

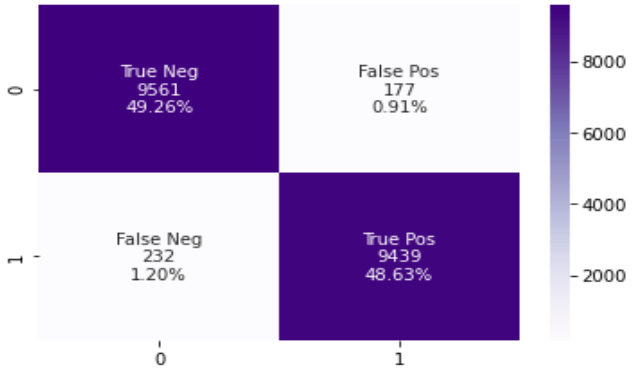


Fig. 9. Confusion Matrix

	precision	recall	f1-score	support
0.0	0.98	0.98	0.98	9738
1.0	0.98	0.98	0.98	9671
accuracy			0.98	19409
macro avg	0.98	0.98	0.98	19409
weighted avg	0.98	0.98	0.98	19409

Fig. 10. Classification report

G. LSVC Machine Learning Algorithm

The confusion matrix provides a comprehensive breakdown of the LSVC model's predictions, distinguishing between true positives, true negatives, false positives, and false negatives.

The high number of true positives (9439) and true negatives (9561) indicates that the LSVC algorithm is highly effective at correctly identifying both cyberthreats and non-threats. The relatively low number of false positives (232) and false negatives (177) suggests that the algorithm has a low error rate in its predictions, minimizing the risk of false alarms and missed threats.

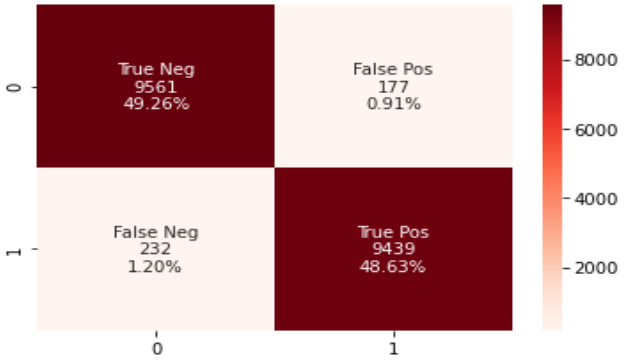


Fig. 11. Confusion Matrix

	precision	recall	f1-score	support
0.0	0.98	0.98	0.98	9738
1.0	0.98	0.98	0.98	9671
accuracy			0.98	19409
macro avg	0.98	0.98	0.98	19409
weighted avg	0.98	0.98	0.98	19409

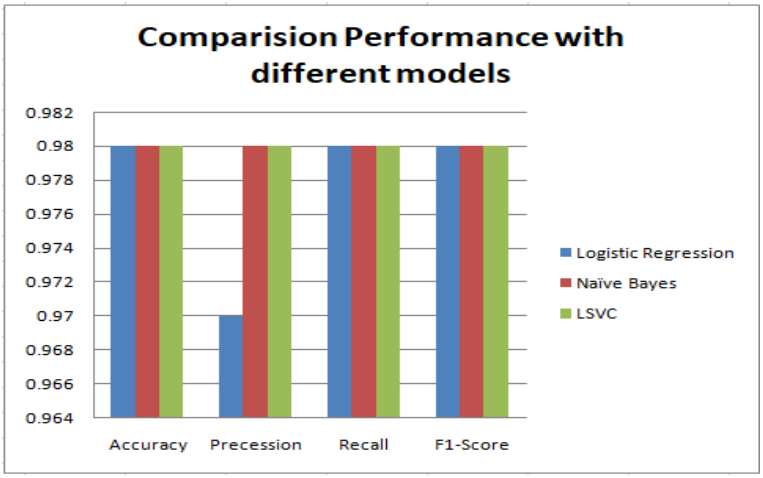
Fig. 12. Implmentation Flow Diagram

H. Comparision Table

Comparison of Performance Metrics with Different Models

S.N	Model/Parameter	Accuracy	Precession	Recall	F1-Score
1	Logistic Regression	0.98	0.97	0.98	0.98
2	Naïve Bayes	0.98	0.98	0.98	0.98
3	LSVC	0.98	0.98	0.98	0.98

I. Performance Comparision with different Machine Learning Models



1) Accuracy*:*
Accuracy measures the proportion of correct predictions (both true positives and true negatives) out of the total number of predictions. All three models achieved an accuracy of 0.98, indicating they correctly classified 98% of the instances.

2) Precision:
Precision is the ratio of true positive predictions to the total number of positive predictions (true positives + false positives). A high precision indicates a low false positive rate. Logistic Regression has a slightly lower precision (0.97) compared to Naïve Bayes and LSVC (both 0.98), suggesting Logistic Regression has a marginally higher rate of false positives.

3) Recall:
Recall is the ratio of true positive predictions to the total number of actual positives (true positives + false negatives). A high recall indicates a low false negative rate. All models achieved a recall of 0.98, meaning they effectively identified 98% of the actual positives.

4) F1-Score:
The F1-Score is the harmonic mean of precision and recall, providing a balanced measure of a model's performance. An F1-Score close to 1 indicates excellent performance. All three models have an F1-Score of 0.98, reflecting their high overall effectiveness in both precision and recall.

6. Conclusion

The comparative analysis of the machine learning models Logistic Regression, Naïve Bayes, and Linear Support Vector Classification (LSVC) demonstrates their efficacy in the context of cyberthreat detection. Each model achieves high performance across the key metrics of Accuracy, Precision, Recall, and F1-Score, all scoring approximately 0.98. These results indicate that the systematic framework employed is highly effective in identifying cyberthreats with minimal error rates. Logistic Regression showed slightly lower precision compared to the other models, suggesting a marginally higher rate of false positives, but still maintains high overall performance. Naïve Bayes and LSVC exhibited almost identical and outstanding performance metrics, highlighting their robustness and reliability in this application. The findings confirm that machine learning models can significantly enhance the accuracy and efficiency of cyberthreat detection systems. These models, when integrated into a comprehensive cybersecurity framework, can help organizations proactively identify and mitigate potential threats, thus strengthening their security posture.

Future Scope
Combining multiple machine learning models into an ensemble approach could enhance prediction accuracy and robustness. Techniques such as stacking, boosting, or bagging might be explored to create a hybrid model that leverages the strengths of individual models.

7. References

1. “SolarWinds hackers linked to known Russian spying tools, investigators say.” 2022. Accessed: Oct. 10, 2022. [Online]. Available: https://cybernews.com/news/solarwinds-hackers-linked-to-known-russianspying- tools-investigators-say/.
2. R. McMillan. “Definition: Threat intelligence.” Accessed: Nov. 10,2022. [Online]. Available: https://gartner.com/ .

3. D. Shackleford, Who's Using Cyberthreat Intelligence and How, SANSI nst., North Bethesda, MD, USA, 2015.
4. H. Dalziel, How to Define and Build an Effective Cyber Threat Intelligence Capability, Syngress, Waltham, MA, USA, 2014.
5. C. Fachkha and M. Debbabi, "Darknet as a source of cyber intelligence: Survey, taxonomy, and characterization," IEEE Commun. Surveys Tuts., vol. 18, no. 2, pp. 1197–1227, 2nd Quart., 2015.
6. J. Robertson et al., Darkweb Cyber Threat Intelligence Mining. Cambridge, U.K.: Cambridge Univ. Press, 2017.
7. W. Tounsi and H. Rais, "A survey on technical threat intelligence in the age of sophisticated cyber attacks," Comput. Security, vol. 72, pp. 212–233, Jan. 2018.
8. T. D. Wagner, K. Mahbub, E. Palomar, and A. E. Abdallah, "Cyber threat intelligence sharing: Survey and research directions," Comput. Security, vol. 87, Nov. 2019, Art. no. 101589.
9. M. S. Abu, S. R. Selamat, A. Ariffin, and R. Yusof, "Cyber threat intelligence—Issue and challenges," Ind. J. Elect. Eng. Comput. Sci., vol. 10, no. 1, pp. 371–379, 2018.
10. A. Ibrahim, D. Thiruvady, J.-G. Schneider, and M. Abdelrazek, "The challenges of leveraging threat intelligence to stop data breaches," Front. Comput. Sci., vol. 2, p. 36, Aug. 2020.
11. M. R. Rahman, R. Mahdavi-Hezaveh, and L. Williams, "What are the attackers doing now? Automating cyber threat intelligence extraction from text on pace with the changing threat landscape: A survey," 2021, arXiv:2109.06808.
12. M. R. Rahman, R. Mahdavi-Hezaveh, and L. Williams, "A literature review on mining cyberthreat intelligence from unstructured texts," in Proc. Int. Conf. Data Min. Workshops (ICDMW), 2020, pp. 516–525.
13. R. Brown and P. Stirparo, SANS 2022 Cyber Threat Intelligence urvey, SANS Inst., North Bethesda, MD, USA, 2022.