Dean&Francis

ISSN 2959-6157

Dynamic Adjustment of the TLS Protocol Based on Naive Bayes

Kaikun Li

Department of Software Engineering, College of Changchun University of Science and Technology in Changchun, China
Email: lfllighy1234@163.com

Abstract:

The adaptive adjustment of the TLS (Transport Layer Security) protocol aims to address various security threats and performance requirements in the current network environment. It is crucial for the reliability of online transactions and data transmissions. This research will adopt a comprehensive research approach, integrating network traffic analysis, naïve bayes machine learning algorithms and simulation experiments. It is expected to develop a mechanism that can dynamically adjust the cipher suite parameters of the TLS protocol according to the real-time network situation, so as to strengthen the security of data transmission and improve the efficiency of network communications.

Keywords: TLS protocol, naive bayes, self-adjustment, cipher suite parameters

1. Introduction

In the digital age, the security of network communications has become the key to safeguarding personal privacy, corporate interests and even national security. As one of the most widely used encryption protocols on the Internet, the performance and security of the Transport Layer Security (TLS) Protocol directly affect the reliability of online transactions and data transmissions. However, with the increasing complexity of the network environment, the TLS Protocol is facing new challenges, such as the frequent need for key updates and diverse network attack means. Therefore, researching the adaptive adjustment of the TLS Protocol is not only an important supplement to the existing network security theories but also has far-reaching guiding significance for practical applications.

To expand the depth and breadth of this research, an exploration of the adaptive adjustment of TLS protocol's performance in diverse network environments such as high latency, high packet loss rate and bandwidth limitations will be conducted. On the technical level, the Naive Bayes algorithm model, which can predict future security threats according to network traffic characteristics and historical data of security events and then adjust the cipher suite parameters of the TLS protocol correspondingly, will be employed in this research. The performance of these models will be evaluated in the actual network environment and compared with the existing security strategies to verify their effectiveness and superiority.

2. Literature Review

A wide variety of studies have been carried out regarding the implementation of prediction with the TLS protocol and Naive Bayes. The following lists some of these studies.

- Asadzadeh Kaljahi +[1] focuses on improving the

SSL/TLS protocol. It puts forward a mechanism called TSSL (Trust Model-based SSL/TLS). The SSL/TLS protocol will be enhanced by using a trust model. In conclusion, The TSSL mechanism proposed in the literature, which improves the SSL/TLS protocol by using the trust model, has achieved certain results in enhancing protocol security and optimizing trust evaluation.

- Zhou, Jiuxing + [2] centered around analyzing the challenges and advances in dealing with TLS 1.3-encrypted traffic. It comprehensively explores various aspects, including how the strengthened encryption features of TLS 1.3 pose difficulties in traffic analysis. In conclusion, it clearly outlines numerous challenges in analyzing TLS 1.3-encrypted traffic and shows innovative approaches developed in terms of traffic analysis methods, which can partly mitigate the challenges.
- Diemert, Denis and Tibor Jager [3] aims to investigate and determine theoretically sound cryptographic parameters for real-world deployments of TLS 1.3. In conclusion, it provides a comprehensive analysis and determination of appropriate cryptographic parameters for TLS 1.3 in real-world deployments based on sound theoretical foundations and demonstrates which specific cryptographic parameters can ensure the tight security of TLS 1.3.
- Fletcher-Lloyd, Nan +[4] focuses on developing a Markov Chain Model for identifying changes in the daily activity patterns of people living with dementia. It involves the process of feature extraction, similar to how feature extraction is needed for the adaptive adjustment of the cipher suite parameters of the TLS protocol. In conclusion, in the TLS protocol aspect, the similarity in the feature extraction processescan assist in determining whether the cipher suite parameters need to be adjusted.
- Matsuda, Koji +[5] aims to comprehensively understand and reveal the characteristics, performance advantages, and disadvantages of different personalized federated learning methods. In conclusion, It has been found that the fine-tuned standard federated learning method shows superiority over personalized federated learning methods in certain situations, which implies that when the Naive Bayes algorithm is applied to adjust TLS cipher suite parameters, fine-tuning the model can also potentially improve its performance.
- Li, Xinyi + [6] develops the Caring framework for achieving collaborative and cross-domain Wi-Fi sensing, especially for human activity recognition, to address the challenge of handling heterogeneous data from different domains to make the Wi-Fi sensing work effectively in a collaborative manner. In conclusion, the proposed Caring framework is effective in handling heterogeneous data from different domains for collaborative Wi-Fi sensing, which helps to make more precise judgments on whether and how to adjust the cipher suite parameters (in the TLS context)

3. Date Source

Set up an environment and simulate different network environments, server configurations and client request scenarios. Use the network packet capture tool Wireshark and performance detection tools to obtain the negotiation data during the TLS handshake process of mobile phones as well as the performance data in the communication process. Some parameters in TLS will be used in this experiment, including TLS Protocol Version, Encryption Algorithm for Key Exchange, Symmetric Cryptographic Algorithm, Message Authentication Algorithm and so on.

4. Naïve Bayes algorithms

Naive Bayes is a remarkably simple yet highly effective probabilistic classification algorithm that is fundamentally based on Bayes' theorem. Bayes' theorem, which holds a central and crucial position in the extensive field of probability theory, is widely recognized and studied for its profound implications and applications in various domains of data analysis and machine learning. It serves as the cornerstone and guiding principle upon which the Naive Bayes algorithm is constructed and operates, enabling it to make predictions and classifications based on the principles of conditional probability and prior knowledge. Its formula is $p(A|B)=\frac{p(B|A)\times p(A)}{P(B)}$, Among them, $p(A|B)$ represents the probability of event A occurring under the condition that event B occurs, which is the posterior probability. $p(A)$ is the prior probability of event A occurring. $p(B|A)$ is the probability of event B occurring under the condition that event A occurs. $p(B)$ is the probability of event B occurring. In the Naive Bayes algorithm for classification problems, the classes are usually regarded as event A, and the features are regarded as event B. Therefore, the target we want to calculate is to calculate the probability of class A under the condition that event B occurs, that is, under the given features, and we need to calculate $p(A|B)$. For samples with m features $X_1, X_2, X_3, \ldots, X_m$ and n categories C_i $(i=1,2,3,\ldots,n)$, to determine which of the n categories C_i $(i=1,2,3,\ldots,n)$ it belongs to, we need to calculate the conditional probability of each category, that is $p(C_i|X_1,X_2,X_3,\ldots,X_m)(i=1,2,3,\ldots,n)$.

ISSN 2959-6157

According to the Naive Bayes formula, $p(C_i \mid X_1, X_2, X_3, \ldots, X_m) = \frac{p(X_1, X_2, X_3, \ldots, X_m \mid C_i) \times p(C_i)}{p(X_1, X_2, X_3, \ldots, X_m)}$, and m features are independent of each other, therefore $p(X_1, X_2, X_3, \ldots, X_m \mid C_i) = p(X_1 \mid C_i) \times p(X_2 \mid C_i) \times p(X_3 \mid C_i) \times \ldots \times p(X_m \mid C_i)$ and $p(X_1, X_2, X_3, \ldots, X_m)$ remains unchanged, therefore $p(C_i \mid X_1, X_2, X_3, \ldots, X_m) \propto p(X_1 \mid C_i) \times p(X_2 \mid C_i) \times p(X_3 \mid C_i) \times \ldots \times p(X_m \mid C_i) \times p(C_i)$.

To determine which category the sample belongs to, compare the magnitudes of each $h(i) = p(C_i) \times \prod_{j=1}^{m} p(X_j \mid C_i) (i = 1, 2, 3, \ldots, n)$, and the probability magnitude that the sample belongs to the i-th category is $p(C_i \mid X_1, X_2, X_3, \ldots, X_m) = \frac{h(i)}{\sum_{i=1}^{n} h(i)}$. For discrete cocnditional probabilities, $p(X_j \mid C_i) = \frac{N(X_j, C_i)}{N(C_i)}$, Among them, $N(C_i)$ represents the quantity size of category $C_i (i = 1, 2, 3, \ldots, n)$, and $N(X_j, C_i)$ represents the quantity size of the simultaneous occurrence of X_j and C_i. For continuous probabilities, $p(X_j \mid C_i) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(X_j - \mu_j)}{2\sigma_j^2}}$, where μ_j is the mean and σ_j is the variance.

5. Implement

5.1 Data collection and preprocessing.

Collect connection records containing the TLS protocol from the network traffic monitoring tool Wireshark. These records contain detailed information about the TLS protocol, which includes Symmetric Cryptographic Algorithm, Encryption Algorithm for Key Exchange, Message Authentication Algorithm, Key Length and TLS Protocol Version, and then count the CPU Utilization Rate, the Type of Client Request and the Network Packet LossRate. At last, conduct security analysis on each connection record through security analysis software, and mark the result obtained with the category to which it belongs, the categories are “insecure”, “secure”, and “at risk”.

Remove invalid or incomplete records. If the cipher suite parameter part lacks key information (such as the absence of an encryption algorithm), that record will be deleted. Meanwhile, check the consistency of the data to ensure that the combinations of encryption algorithms and key lengths are reasonable and avoid illogical parameter combinations (for example, an AES key with a length of 16 bits, while the actual minimum key length of AES is usually 128 bits).

Determine the proportions of the training set, validation set and test set according to the total amount of data and the actual situation. The proportion is 70% for the training set, 15% for the validation set, and 15% for the test set. For each security level category, extract data according to the determined proportion of the training set. Similarly, by means of stratified sampling, for each security level category, extract data according to the proportions of the validation set and the test set. The validation set is mainly used to adjust the hyperparameters of the model during the model training process in order to optimize the model’s performance. The test set is used to evaluate the final performance of the model after the model training is completed, such as metrics like accuracy, recall and so on. After the division is completed, perform encoding processing on each feature.

5.2 Use the Naive Bayes model to train data.

The security category: $C = \{ C_1, C_2, C_3 \}$. According to this, calculate the prior probability: $p(C_i) = \frac{N(C_i)}{N} (i = 1, 2, 3)$, Among them, $N(C_i)$ is the number of training samples belonging to the category, and N is the total number of samples in the training set. For each feature X_j, calculate the conditional probability $p(X_j \mid C_i)$ under each category C_i For discrete features, obtain the conditional probability by counting the occurrence frequencies of the features in each category. For continuous features, they follow a normal distribution (Gaussian distribution) under the category, and $p(X_j \mid C_i) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(X_j - \mu_j)}{2\sigma_j^2}}$ will be calculated, where μ_j is the mean, and σ_j is the variance. After the above calculations, once the prior probability $p(C_i)$ and the conditional probability $p(X_j \mid C_i)$ are obtained, the Naive Bayes model is basically constructed. These probability values will be used for the subsequent classification and adaptive adjustment of new TLS cipher suite parameters. Among them, the classification method is to compare the magnitudes of each

$h(i) = p(C_i) \times \prod_{j=1}^{m} p(X_j \mid C_i)(i = 1,2,3)$, find the largest one $h(k)$, and classify the sample into C_k.

5.3 Model evaluation

The Confusion Matrix is a particular kind of table that is specifically used to evaluate the performance of a classification model. It clearly shows the specific relationship between the model's prediction results and the actual labels. The security category is $C = \{C_1, C_2, C_3\}$, which have three categories, Thus, its confusion matrix is a 3×3 matrix, as shown in the Table of Explanation of Confusion Matrix:

Table 1 Explanation of Confusion Matrix

Actual classification / Predicted classification	C_1	C_2	C_3
C_1	TPC_1	FPC_{1-2}	FPC_{1-3}
C_2	FPC_{2-1}	TPC_2	FPC_{2-3}
C_3	FPC_{3-1}	FPC_{3-2}	TPC_3

Among them, $TPC_i (i = 1,2,3)$ represents the true positive cases, that is, for category i, the number of samples that are actually of category i and are also correctly predicted as category i by the model. FPC_{i-j} represents the false positive cases, that is, for categories i and $j (i \neq j)$, the number of samples that are actually of category i but are wrongly predicted as category j by the model.

Accuracy refers to the specific proportion of samples that are precisely and correctly classified by the model to the overall total number of samples. $Accuracy = \frac{\sum_{i=1}^{3} TPC_i}{\sum_{i=1}^{3}\sum_{j=1}^{3} M_{ij}}$, where M_{ij} refers to the element in the $i-th$ row and $j-th$ column of the confusion matrix.

Precision is the particular proportion of samples that are truly and actually of a certain specific category among those which are predicted as that very same category by the model. $Precision_i = \frac{TPC_i}{\sum_{j=1}^{3} M_{ij}}$.

Recall is the significant proportion of samples that are accurately and correctly classified as a particular certain category by the model precisely when they are truly and actually of that very same specific category. $Recall_i = \frac{TPC_i}{\sum_{j=1}^{3} M_{ji}}$.

The F1-score is the harmonic mean of precision and recall. For category i, $F1-score_i = \frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i}$.

It takes both precision and recall into comprehensive consideration and can evaluate the performance of the model in classifying each category more comprehensively.

5.4 TLS Protocol Implementation Model

5.4.1 Client integration

On the TLS client side, a connection request is initiated and relevant features are extracted. The features which are the same as those for model training will be input into the trained Naive Bayes model, and outputs the predicted security category result of "secure", "insecure" or "at risk". Based on this, the client takes corresponding actions. If the connection is predicted to be at risk or insecure, the client can terminate the request or prompt the user and ask whether to continue, along with providing risk information.

5.4.2 Server integrationthe

On the server side, once a TLS connection request arrives from a client, relevant features, identical to those used in model training, are carefully extracted. These features are then fed into the model for prediction. Based on the outcome, the server makes a determination. If the prediction is "insecure", the request is flatly rejected. In case the prediction is "at risk", additional security measures such as enhanced authentication are implemented to safeguard the connection.

5.5 Monitoring and Updating the Model

5.5.1 Model Monitoring

In order to comprehensively assess the performance of the model, it is necessary to calculate the AUC of the model in the real environment on a weekly basis. Essentially, the closer the AUC value is to 1, the more powerful and

ISSN 2959-6157

reliable the discriminative ability of the model turns out to be. If, over an extended period, the AUC value exhibits a significant and continuous downward trend, it clearly signals that the overall performance of the model is gradually deteriorating. Additionally, it is crucial to closely and continuously monitor both the True Positive Rate (TPR) and the False Positive Rate (FPR). A substantial and sudden drop in TPR truly implies that the model fails to detect and misses a large number of connections that are actually and truly at risk. Conversely, a sharp and abrupt rise in FPR evidently indicates that the model wrongly misjudges a great many secure connections as being at risk. Both of these scenarios can have a profound and negative impact on the proper and normal operation of the TLS protocol, potentially leading to security vulnerabilities and disruptions in network communications.

5.5.2 Update the model

Analyze each feature's importance in the model weekly. This assessment is accomplished by precisely computing the contribution of its conditional probability to the overall prediction. In the event that the influence of a crucial feature diminishes, it may potentially signify either an update in the TLS protocol or a shift in attack methodologies. Consequently, it becomes necessary to contemplate making adjustments to the feature set. Moreover, with the continuous evolution of TLS and the emergence of novel threats, new pertinent features are likely to surface. If these newly emerged features have an impact on connection security, they should be incorporated into the model for further training and evaluation, thereby ensuring the model's adaptability and effectiveness in safeguarding the network environment.

6. Environment and System

6.1 Programming Languages and Development Frameworks

The highly versatile and widely adopted Python programming language is comprehensively and effectively used to implement the Naive Bayes algorithm and conduct extensive data processing. It has useful libraries like NumPy for numerical calc., Pandas for data proc. and Scikit-learn with Naive Bayes classifier. These remarkable libraries, with their diverse functionalities and features, simplify the often complex and intricate development process and improve the overall development efficiency, making Python a good option.

For the crucial tasks of data visualization and result presentation, Matplotlib and Seaborn libraries are used in combination. They can draw intuitive and appealing charts like confusion matrix diagrams and precision-recall curves, which help analyze the model's performance in predictive capabilities and TLS security classification results, important for evaluating the security assessment system.

6.2 Data Acquisition Tools

The network traffic monitoring tool Wireshark is specifically and effectively used to capture TLS traffic and extract detailed and crucial cipher suite parameter information. Its advanced functions enable setting accurate filters for TLS-related packets, improving data collection efficiency. Also, it can save the captured data in a convenient format like PCAP files, which is beneficial for subsequent processing and analysis.

7. Result Analysis

7.1 Model Performance Evaluation

In order to evaluate the performance of the Naive Bayes classifier, 200 data records are used for training and testing. Among them, the training set accounts for 70%, that is, 140 data records; and the test set accounts for 30%, that is, 60 data records. The analysis results are shown in the confusion matrix in Table of Confusion Matrix:

Confusion Matrix

Actual / prediction	1(insecure)	2(secure)	3(at risk)
1(insecure)	12	0	0
2(secure)	0	33	1
3(at risk)	0	0	14

Among them, the number of samples predicted to be insecure is 12, and the number of actually insecure samples is also 12. Therefore, $TPC_1 = 12$, $FPC_{1-2} = 0$ and $FPC_{1-3} = 0$. The number of samples predicted to be secure is 33, while the number of actually secure samples is 34. There is 1 sample that is actually secure but predicted to be at risk. So, $TPC_2 = 33$, $FPC_{2-1} = 0$ and $FPC_{2-3} = 1$. The number of samples predicted to be at risk is 14, and the number of actually at−risk samples is also 14. Hence, $TPC_3 = 14$, $FPC_{3-1} = 0$ and $FPC_{3-2} = 0$. According to the

formula,

$$Accuracy = \frac{\sum_{i=1}^{3} TPC_i}{\sum_{i=1}^{3}\sum_{j=1}^{3} M_{ij}} = \frac{12+33+14}{60} \times 100\% = 98.33\%,$$

$$Precision_1 = \frac{TPC_1}{\sum_{j=1}^{3} M_{1j}} = \frac{12}{12+0+0} \times 100\% = 1,$$

$$Precision_2 = \frac{TPC_2}{\sum_{j=1}^{3} M_{2j}} = \frac{33}{0+33+1} \times 100\% = 97.06\%,$$

$$Precision_3 = \frac{TPC_3}{\sum_{j=1}^{3} M_{3j}} = \frac{14}{0+0+14} \times 100\% = 1,$$

$$Recall_1 = \frac{TPC_1}{\sum_{j=1}^{3} M_{j1}} = \frac{12}{12+0+0} \times 100\% = 1,$$

$$Recall_2 = \frac{TPC_2}{\sum_{j=1}^{3} M_{j2}} = \frac{33}{33+0+0} \times 100\% = 1,$$

$$Recall_3 = \frac{TPC_3}{\sum_{j=1}^{3} M_{j3}} = \frac{14}{14+1+0} \times 100\% = 93.33\%,$$

$$F1-score_1 = \frac{2 \times Precision_1 \times Recall_1}{Precision_1 + Recall_1} = \frac{2 \times 1 \times 1}{1+1} \times 100\% = 1,$$

$$F1-score_2 = \frac{2 \times Precision_2 \times Recall_2}{Precision_2 + Recall_2} = \frac{2 \times 0.9706 \times 1}{0.9706+1} \times 100\%$$

$$= 98.51\%, F1-score_3 = \frac{2 \times Precision_3 \times Recall_3}{Precision_3 + Recall_3}$$

$$= \frac{2 \times 1 \times 0.9333}{0.9333+1} \times 100\% = 96.55\%$$

As shocwn in the Table of Modeling Evaluation.

Table 2 Modeling Evaluation

	1(insecure)	2(secure)	3(at risk)
Precision	1.0	97.06%	1.0
Recall	1.0	1.0	93.33%
F1−score	1.0	98.51%	96.55%

7.2 Permutation Vector Feature Contribution Degree

The feature contribution degree helps to enable a more accurate and detailed understanding of the significance and influence of each individual feature within the context of the Naive Bayes algorithm and the overall classification task.

As shown in Figure of Permutation Importance, the feature contribution degree of the feature vector to the dependent variable is judged through 20 permutation tests.

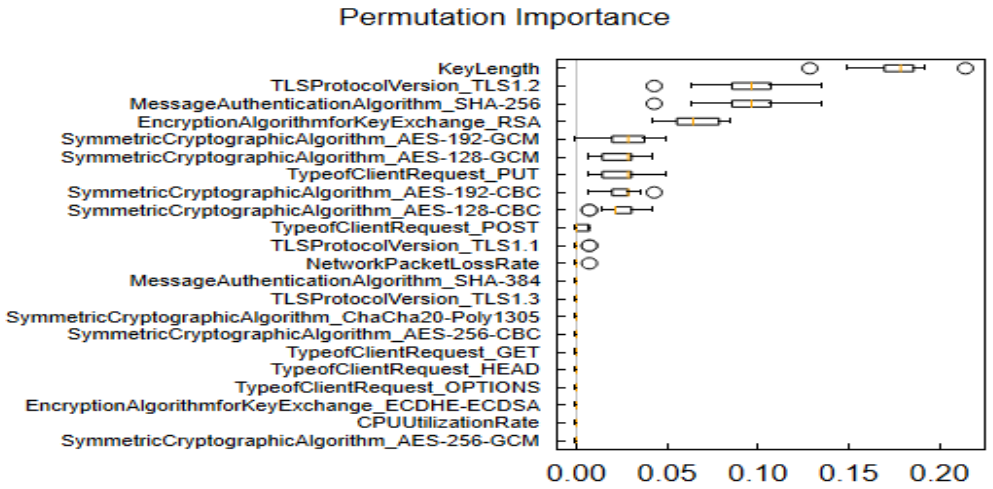

Figure 1 Permutation Importance

From the perspective of the magnitude of feature contribution degree, the key length has a relatively high importance, which indicates that in the fields of network security and data encryption, the key length has a significant impact on the overall security and system performance. Generally, a longer key can provide higher security, but it may also lead to the problem of increased consumption of computational resources. The TLS version number also has a certain level of importance. Different TLS versions vary in terms of security and performance. Newer ver-

ISSN 2959-6157

sions like TLS 1.3 usually offer stronger encryption and a more efficient handshake process. The message authentication algorithm and the symmetric encryption algorithm show relatively high importance in the graph. These algorithms are used to ensure the integrity and confidentiality of data and are crucial components of network security. The network packet loss rate and CPU usage have relatively lower importance in the graph, but they still cannot be ignored. The network packet loss rate affects the reliability of data transmission, while the CPU usage reflects the load situation of the system and has an impact on the overall performance of the system.

7.3 ROC curve

The ROC (Receiver Operating Characteristic) curve of this experiment is shown in Figure of ROC curve.

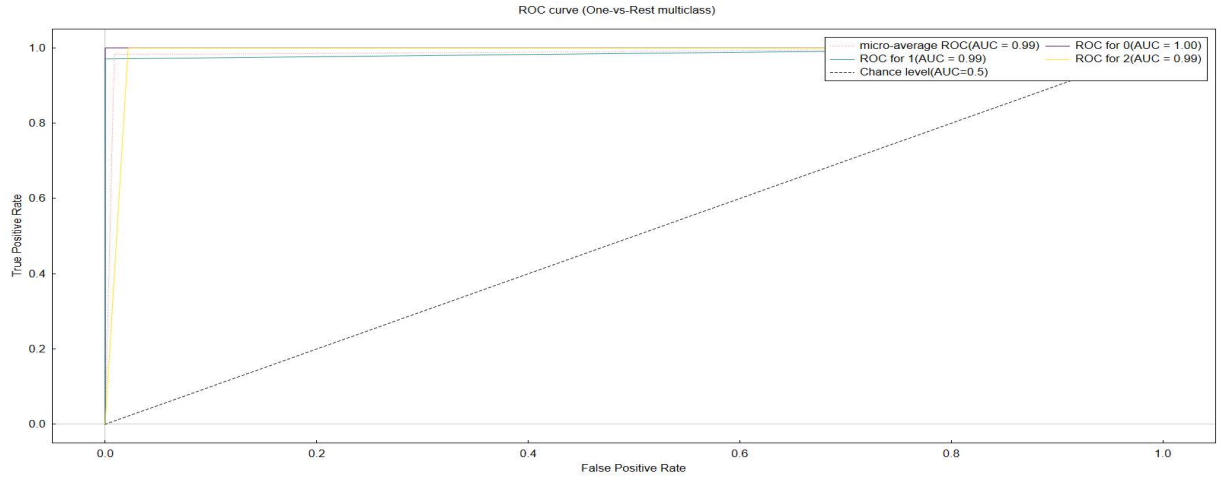

Figure 2 ROC curve

As can be seen from the ROC curve graph, the ROC curve of the "insecure" category performs extremely well, with an Area Under the Curve (AUC) of 1.00. This implies that when distinguishing the "insecure" category from other categories, the model has a perfect discriminative ability. The ROC curves of the "secure" and "at risk" categories almost overlap, and both have an AUC value of 0.99. This indicates that the model also has a very high accuracy when distinguishing these two categories from other categories. Although it does not reach the perfect level of the "insecure" category, the AUC value of 0.99 already shows that the model has a strong discriminative ability in these two categories. The ROC curve at the random level (dashed line) has an AUC value of 0.5. The ROC curves of all categories are far above the random level, which further proves the effectiveness of the model.

8. Conclusion

This paper focused on the multi-class classification of TLS cipher suite parameters which is crucial for network security and integrity. It collected a large amount of data from various actual network environments with complex and dynamic security features and carefully designed simulated scenarios. Key parameters such as encryption algorithm types, key exchange protocols, message authentication code algorithms, and protocol versions were screened and organized. After data preprocessing, a classification model based on the Naive Bayes algorithm was constructed by computing prior and conditional probabilities in strict accordance with the algorithm's principle. In conclusion, the Naive Bayes-based model can efficiently and accurately classify novel TLS parameters into corresponding security categories based on posterior probabilities. It can, to a large extent, accurately identify cipher suites with different security levels, providing a highly valuable reference for network security assessment and management, which is essential for maintaining a secure and reliable network infrastructure.

References

[1]Asadzadeh Kaljahi, Maryam, Ali Payandeh, and Mohammad Bagher Ghaznavi-Ghoushchi. "TSSL: improving SSL/TLS protocol by trust model." Security and Communication Networks 8.9 (2015): 1659-1671.

[2]Zhou, Jiuxing, et al. "Challenges and Advances in Analyzing TLS 1.3-Encrypted Traffic: A Comprehensive Survey." Electronics 13.20 (2024): 4000.

[3]Diemert, Denis, and Tibor Jager. "On the tight security of TLS 1.3: Theoretically sound cryptographic parameters for real-world deployments." Journal of Cryptology 34.3 (2021): 30.

[4]Fletcher-Lloyd, Nan, et al. "A Markov Chain Model for

Identifying Changes in Daily Activity Patterns of People Living with Dementia." IEEE Internet of Things Journal (2023).
[5]Matsuda, Koji, et al. "Benchmark for Personalized Federated Learning." IEEE Open Journal of the Computer Society (2023).
[6]Li, Xinyi, et al. ": Towards Collaborative and Cross-Domain Wi-Fi Sensing: A Case Study for Human Activity Recognition." IEEE Transactions on Mobile Computing 23.2 (2023): 1674-1688.
[7]Pajila, PJ Beslin, et al. "A comprehensive survey on naive bayes algorithm: Advantages, limitations and applications." 2023 4th International Conference on Smart Electronics and Communication (ICOSEC). IEEE, 2023.
[8]Kohavi, Ron. "Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid." Kdd. Vol. 96. 1996.
[9]Lachiche, Nicolas, and Peter A. Flach. "Improving accuracy and cost of two-class and multi-class probabilistic classifiers using ROC curves." Proceedings of the 20th international conference on machine learning (ICML-03). 2023.
[10]Ontivero-Ortega, Marlis, et al. "Fast Gaussian Naïve Bayes for searchlight classification analysis." Neuroimage 163 (2017): 471-479.