Computer Science and Information Systems 22(1):79–103 https://doi.org/10.2298/CSIS240728069F

Anomalous Traffic Identification Method for POST Messages Based on Gambling Website Templates

Zhimin Feng[1], Dezhi Han[1], Songyang Wu[2,⋆], Wenqi Sun[2], and Shuxin Shi[1]

[1] College of information Engineering, Shanghai Maritime University
201306 Shanghai, China
fengzhimin@stu.shmtu.edu.cn
dzhan@shmtu.edu.cn
shishuxin@stu.shmtu.edu.cn

[2] Network Security Center, The Third Research Institute of
the Ministry of Public Security
200031 Shanghai, China
wusongyang@stars.org.cn
sunwenqi@gass.ac.cn

Abstract. Malicious websites pose significant social risks, necessitating automatic, efficient, and accurate identification methods. This paper proposes a POST traffic classification method based on website templates to identify abnormal traffic from gambling websites. Using Fiddler, POST message data is collected from several gambling sites, extracting features like URLs, cookie parameters, and request body parameters to create a Gambling Website Single POST Message Dataset (GSPD). These features are converted into vector representations with Word2Vec and TF-IDF techniques. Hierarchical clustering identifies template-generated types, achieving unsupervised template recognition. Using clustering results, individual POST messages are labeled and features are extracted using TF-IDF and mutual information methods. The parameters of a Support Vector Machine (SVM) are then optimized with the Particle Swarm Optimization (PSO) algorithm for optimal classification. Experimental results show the model's excellent performance, with a test set accuracy of 0.9985 and high precision, recall, and F1-scores, effectively identifying gambling and other illegal websites.

Keywords: Template recognition, Illegal Website Detectio, feature extraction, POST traffic classification.

1. Introduction

The Internet has become the primary source of information for people today. However, it is also inundated with malicious content, particularly gambling websites, which are closely linked to cybercrime and pose significant harm to society. Statistics from 2023 reveal that 26% of the global population is involved in online gambling. Manual identification is impractical because of the vast number of gambling websites and their continual updates. Consequently, there is an urgent need to develop an automatic, efficient, and accurate method for identifying abnormal traffic from gambling websites.

[⋆] Corresponding author

Current methods for identifying gambling websites can be categorized into several groups: Blocklists, URLs, web content, and hybrid features. Blocklisting methods, which rely on collecting malicious URLs or domain names, are helpful but costly to maintain and inefficient. URL-based methods[5] classify websites by extracting features from URLs, but their accuracy is limited due to the minimal URL information. Web content methods[25,16,17,15,8,7,23,14] analyze content features such as HTML text, images, links, and JavaScript code to perform recognition. Hybrid feature methods [3,22,4,1,9,24,21,26] combine multiple features to enhance classification accuracy. Currently, in the field of cybersecurity, deep learning has been proven to be effective in improving the detection of complex traffic[11,12,13], which provides strong support for the optimisation of gambling site identification techniques . In addition, the combination of federated learning and blockchain technology opens up new ways for the construction of distributed and privacy-preserving gambling site detection systems[10], thus enhancing the security and robustness of the system . And for the problem of inconsistent data distribution, the latest research proposes a feedback semi-supervised learning method[2], which shows better robustness in dealing with highly polluted data, which is important for improving the detection accuracy of gambling website traffic.However, web content and hybrid feature approaches struggle with accurately recognizing gambling websites due to the complexity of dealing with images.

While researching gambling websites, this paper finds that most gambling website operators use site-building script technology[18] to automatically generate websites to reduce costs and simplify creation and management. By analyzing the HTTP POST behavioural data submitted by users, this study identifies similar characteristics in the HTTP POST data of gambling websites generated using the same template. It is observed that the cookie parameters in the request headers of these sites are remarkably similar. While some cookie parameter names are commonly used for specific purposes, such as authentication and session management (e.g., session, auth_token), these names are not mandatory, and web developers can choose names freely, For example, the parameter ‘isAutoPay’ is used to identify whether a user has auto-pay enabled, and the parameter ‘hasPhone’ is used to identify whether a user has bound or verified a mobile phone number. Furthermore, the analysis reveals that web developers specify many cookie parameter names rather than being strictly defined. Additionally, there is a high degree of similarity in the URL segments following the domain name for the same behaviours on gambling websites generated under the same template in POST messages. The parameters in the request body also exhibit a significant level of similarity.

Based on the HTTP POST data, this paper employs a pre-trained Word2Vec model, converting the data into vector representations using a TF-IDF weighted average, and applies hierarchical clustering to group websites generated from the same template into a single class. While this method can identify gambling website templates, it is insufficient for recognizing anomalous traffic based on the overall website behaviour. Therefore, this paper uses templates to classify individual POST messages. The process involves TF-IDF feature vectorization, feature re-extraction using the mutual information method, and classification using a support vector machine (SVM) optimized with the particle swarm algorithm (PSO) to find the optimal model parameters. The main contributions of this paper are as follows:

1. In this paper, we utilize the Fiddler tool to capture POST messages associated with various behaviours such as login, registration, adding bank cards, and betting on gambling websites that use existing templates. We process the contents of their request headers, request lines, and request bodies to create a Gambling Website Single POST Message Dataset (GSPD). To facilitate the clustering of gambling websites, we then integrate the POST data from the same websites to form a Gambling Website POST Message Merge Dataset (GPMD).
2. In this paper, by analyzing user-submitted HTTP POST behavioural data, we find that gambling websites generated by the same template exhibit high similarity in cookie parameters within the request headers, parameters in the request body, and word similarity following the URL domain name. Feature pattern discovery using unsupervised clustering. The pre-trained Word2Vec model converts the POST messages into vector representations using a TF-IDF weighted average. The hierarchical clustering technique in unsupervised clustering is applied to cluster the gambling websites generated from the same template.
3. In this paper, we develop a template-based classification method for POST messages that employs TF-IDF feature vectorization and the mutual information method for feature extraction. When classifying a single POST message, we use Support Vector Machines (SVMs) for classification and Particle Swarm Optimization (PSO) to optimize the model parameters, enhancing classification accuracy. This method effectively handles large-scale datasets, quickly and accurately identifies gambling sites, reduces the cost of human intervention, and demonstrates strong adaptability and robustness in dealing with constantly evolving gambling sites.

The remainder of this paper is organized as follows: Section 2 introduces related work on identifying gambling websites. Section 3 details the POST message classification process for the dataset used in this study, including the dataset's creation, the clustering of gambling websites, and the method for classifying individual POST messages. Section 4 validates the proposed method through experiments. Finally, Section 5 concludes the paper and suggests directions for future research.

2. Related Work

Wang et al.[20] classified web page screenshots by integrating visual features and textual content. They extracted visual features using a fine-tuned pre-trained ResNet 34 model, employed OCR (Optical Character Recognition) techniques for text extraction from images, and utilized a Bi-LSTM model for semantic feature extraction from the text. These features were fused using a self-attention mechanism and integrated using a post-fusion method for classification. Chen et al.[4] introduced PG-VTDM (visual and textual content using a decision mechanism), an automated detection system for identifying pornographic and gambling websites based on visual and textual content. The system utilized Doc2Vec to learn textual features from HTML source codes, represented visual features of web page screenshots with a modified bag of visual words (BoVW) algorithm incorporating local spatial relationships of feature points, and trained text and image classifiers. A logistic regression-based data fusion algorithm was designed to integrate the classification results from both features. Sun et al.[16] proposed CT-GDNC (certificated and textual analysis-based classification), a method for classifying gambling domain names (GDNs). This method enhanced classification accuracy through BERT fine-tuning using 10,000 benign data samples from GDNs and the Alexa Top 1 million list. Wang et al.[19] developed a co-training approach combining visual and semantic features of web screenshots to identify gambling websites. They utilized OCR for text extraction, trained CNN and TextRNN classifiers separately, and used a co-training algorithm to retrain unlabeled data. Gu et al.[6] introduced HeCGamb (Heterogeneous Communication Graph-based method to enhance the Gambling app detection performance), a method for detecting mobile gambling applications through encrypted traffic analysis. They analyzed communication features of 175 popular gambling apps in China, identified server domain randomization and inter-application family features, modelled inter-flow relationships from traffic streams, and constructed inter-application relationships for application-level family feature extraction.

In contrast to Wang et al.[20] and Chen et al.[4], whose methods rely on visual features and textual content extracted from webpage screenshots, the approach in this paper starts directly from user-submitted HTTP POST behavioural data. This method captures specific user interaction behaviours by analyzing the cookie parameters in the request headers, the parameters in the request body, and the word features following the URL domain name in the POST messages. HTTP POST messages directly reflect user actions on a website, eliminating the need for intermediate steps involving screenshots and text content. This approach is more efficient and real-time, leveraging the more straightforward structure and reduced noise of POST messages to enhance recognition accuracy.

Unlike the multimodal data fusion methods of Wang et al. [19], which combined images and text, and Sun et al.[16], which also utilized images and text, the method proposed in this paper focuses solely on single-modality POST data. This approach simplifies the model complexity and reduces the time and resource consumption for data processing. Despite this focus on a single modality, the method in this paper achieves exceptionally high recognition accuracy on unimodal data through deep mining and optimization of POST data.

In contrast to other methods, the approach in this paper excels in meticulously analyzing and detecting specific operational behaviours of gambling websites, thereby offering comprehensive monitoring and protection. Compared to Gu et al.[6], whose method relies on complex multi-view semantic information fusion and graph analysis, the method presented here is more concise and efficient in feature extraction and classification. It processes large-scale datasets quickly and accurately, significantly enhancing recognition efficiency and reducing the need for human intervention.

3. Method

This paper presents a method for identifying POST requests related to gambling website traffic, illustrated in Fig. 1 (Overall structure diagram). Firstly, the paper preprocesses the collected POST request data by extracting features from request lines, request bodies, and cookies. For request lines, the complete URL is extracted and segmented into words. For cookies and request bodies, all parameters are extracted. Subsequently, the paper compiles and stores each website's URL vocabulary, cookie parameters, and body parameters from all associated POST requests into a file.

Next, the paper uses the extracted URL vocabulary, cookie, and body parameters to train the Word2Vec model. The TF-IDF weighted average transforms Each website's data into vector representations. Subsequently, hierarchical clustering is conducted using the Ward linkage method to categorize the websites, where each cluster corresponds to a gambling website template. Finally, the clustering results are added to the original data, the Gambling Site Single POST Message Dataset (GSPD), as class labels for each POST request.

Then, this paper proceeds to classify each POST request based on its request line, cookie parameters, and request body parameters. Initially, the paper applies the mutual information method to select features most pertinent to the assigned labels. Subsequently, the Particle Swarm Algorithm (PSO) is employed to optimize the hyperparameters of the support vector machine (SVM) for optimal classification performance. Next, feature unions merge and process the selected features through TF-IDF vectorization. Finally, the paper trains the support vector machine model using the optimized parameters.

This paper will provide detailed descriptions of the dataset production, data vector representation, clustering of gambling sites, and classification based on POST messages in the subsequent subsections.

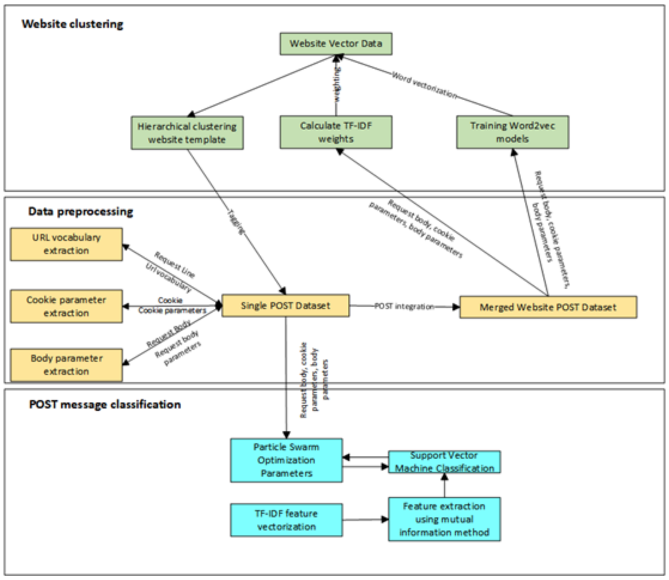
Website clustering
Website Vector Data
weighting
Word vectorization
Hierarchical clustering website template
Calculate TF-IDF weights
Training Word2vec models
Data preprocessing
Tagging
Request body, cookie parameters, body parameters
Request body, cookie parameters, body parameters
URL vocabulary extraction
Request Line
Url vocabulary
Cookie parameter extraction
Cookie
Cookie parameters
Single POST Dataset
POST integration
Merged Website POST Dataset
Body parameter extraction
Request Body
Request body parameters
Request body, cookie parameters, body parameters
POST message classification
Particle Swarm Optimization Parameters
Support Vector Machine Classification
TF-IDF feature vectorization
Feature extraction using mutual information method

Fig. 1. The overall flowchart outlines the three main parts of achieving the final classification in this thesis, including data preprocessing, website clustering, and single-article POST classification. It also specifically refines the detailed processing of each part

3.1. Data set production

HTTP POST message. For a website, common HTTP behaviours include GET and POST requests. GET requests retrieve information from the server, while POST requests are used to submit data to a specified resource, altering the server's state upon execution. Parameters in a GET request are usually appended to the URL as part of the query string. Because URL lengths are limited, GET requests can pass a relatively small amount of data, and POST requests include the data in the request's body rather than in the URL. A POST request can pass significantly since the request body can contain a large amount of data. Hence, this paper selects HTTP POST as the basis for detection. An HTTP POST message typically comprises the following components.

1. The Request Line in an HTTP message follows the ”HTTP method URL HTTP version” format. The HTTP method indicates the action the client intends to perform, such as GET for retrieving resources from the server, POST for sending data to the server, PUT for replacing a resource on the server, DELETE for requesting deletion of a server resource, and other methods like HEAD, OPTIONS, PATCH, and TRACE. The URL (Uniform et al.) specifies the target resource of the request, typically including the protocol (e.g., HTTP, HTTPS), the domain name (e.g., www.example.com), path (e.g., /path/to/resource), and optional query parameters (e.g., ?key1=value1&key2=value2). The HTTP version specifies the version of HTTP being used, commonly HTTP/1.1, HTTP/2, etc.
2. Request Header: The request header consists of multiple lines of key-value pairs, each line of ”key: value” The request header field conveys some of the client's configuration information and the nature of the request so that the server can correctly understand and process the request. The following is the meaning of some of the keys in the request header: Host specifies the domain name of the target server for the request, and Connection indicates that the client wishes to maintain a connection with the server in order to send subsequent requests. Content-Length specifies the length of the request body in bytes. sec-ch-ua indicates the client's user-agent information, including the browser and version. Accept specifies the type of content that the client can handle. Content-Type specifies the media type of the request body. The referer specifies the URL of the facet that initiated the request. Accept-Encoding specifies the content encoding format supported by the client. Accept-Language specifies the client's preferred language. A cookie contains all the cookies the client previously stored on the same server. A cookie contains all cookies stored by the client before the same server.
3. Request Body: This section is located at the end of the HTTP request message, immediately after the request headers. Its structure is simply a piece of data in the format specified by the Content-Type in the request header. The request body is usually closely related to the request line (containing the request method, URL, and HTTP version) and the request headers, but the actual data content is stored entirely in the request body. The request body is the core of the data transfer and can carry complex, structured data in various formats (e.g., form data, JSON, XML, etc.). The data is usually in JSON format if the Content-Type is application/JSON.

GSPD dataset production. By examining the POST messages of the captured gambling websites, this paper finds that for gambling websites that are in the same template constructed by the same template, there is a significant similarity in the URLs, the cookies in the request header, and the body of the request and that the URLs of these websites contain the exact words, and similar parameters and values are passed under the same behaviour. Cookie parameters in the request headers of gambling websites created from the same template tend to be similar. However, some of the parameter names in the cookies are widely received and used for specific purposes, with common authentication cookie names such as session, auth_token, and so on commonly used for authentication, session management, and so on. These are not mandatory specifications; web developers can choose the parameter names. According to the analysis of the data captured in this paper, many of the cookie parameter names are specified by the web developers themselves rather than being strictly regulated. In addition, according to the data analysed in this paper, there is a high degree of similarity in the request body parameters of gambling websites created from the same template. The following is a treatment of the above three components.

1. For the request line, the complete URL is extracted and split into words; when splitting the URL, this paper extracts words from the part of the URL after the domain name and also extracts words from the keys and values in its query parameters.
2. For cookies in the request header, extract the parameters in the cookie, i.e. the part to the left of the equals sign, and do not process the values.
3. For the request body, extract its parameters; if the request body is in the form of ”parameter=value”, extract the part to the left of the equals sign; if the request body is in the form of ”key: value”, extract the critical part.

Fiddler is a robust network debugging tool mainly used to capture and analyse HTTP and HTTPS traffic. It intercepts all network requests and responses from a client (such as a browser or application) to a server and helps users view, modify and debug network data. In this paper, we utilize the Fiddler tool to capture POST message information from various templates of multi-class gambling websites. We segment and compile data from request headers, request lines, and request bodies, including URLs associated with website names, POST behaviours, cookies, and request body parameters. The processed results, including segmented URL words, cookie parameters, and request body parameters, are saved into a file to form the Gambling Site Single POST Message Dataset (GSPD).

GPMD dataset integration. In the data captured for this paper, each gambling website involves multiple POST messages related to actions such as registration, login, adding bank cards, topping up, and betting. Given the focus on website templates for classification and single POST messages for identifying gambling traffic, analysing the features across all POST messages within each gambling website is essential. Based on the Gambling Site Single POST Message Dataset (GSPD), this paper analyses and finds that URL vocabulary information, cookie parameters and parameters in Request Body can be used for website feature pattern discovery. To facilitate this analysis, the paper extracts and compiles URL vocabulary, cookie parameters, and request body parameters from the POST messages obtained from each website. These features are then saved into a file to create a comprehensive Website Merge Dataset (GPMD). Unlike the GSPD, which focuses on the individual behaviours of each gambling site, the GPMD consolidates data from a single website, including only the URL vocabulary, cookie parameters, and request body parameters from the GSPD. Therefore, the Gambling Site POST Message Merge Dataset (GPMD) exclusively includes URL vocabulary, cookie parameters, and request body parameters extracted from the Gambling Site Single POST Dataset (GSPD).

3.2. Data Vector Representation

Word2Vec. Word2Vec is a prominent method for converting natural language into distributed vector representations. It effectively captures word relationships within a multidimensional space and is a crucial preprocessing step in predictive modelling, semantic analysis, and information retrieval tasks. Fig. 2 (Word2vec models) illustrates the Word2Vec process, which includes two primary components: a continuous bag of words (CBOW) and a skip-gram. The CBOW component predicts the target word based on the surrounding context words, whereas the skip-gram component predicts the context words given an input word. These components enable Word2Vec to generate meaningful vector embeddings that reflect semantic relationships and contextual similarities among words in text data.

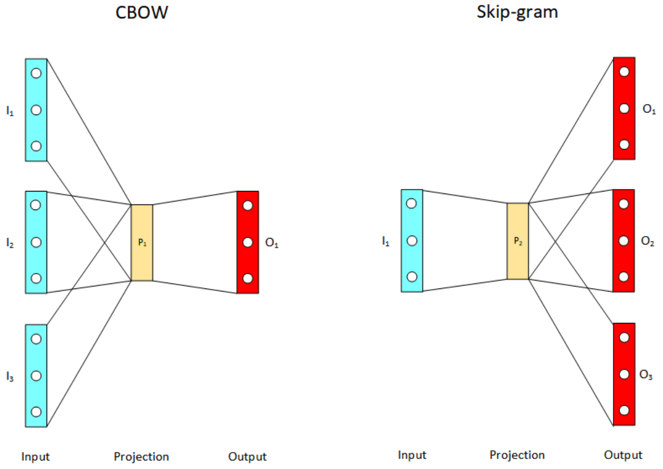

Fig. 2. Word2vec models (including CBOW, Skip-gram)

The Skip-gram component used in this paper is the Skip-gram model, which predicts the context word given the target word, calculates the conditional probability using the softmax function, and trains the word vector by maximising this probability.

In the Skip-gram model, Suppose we have a target word w_t and a window size c,Then the goal of the Skip-gram model is to pass the w_t and its contextual words w_{t+j},The goal of the Skip-gram model is to maximise the following conditional probabilities:

$$P(w_{t+j} \mid w_t)\,. \tag{1}$$

On the corpus as a whole, this goal can be expressed as follows:

$$\prod_{t=1}^{T} \prod_{-c \leq j \leq c, j \neq 0} P\left(w_{t+j} | w_t\right). \tag{2}$$

Where T is the total number of words in the corpus and c is the window size

To calculate this conditional probability, the softmax function is applied:

$$P\left(w_O | w_I\right) = \frac{exp\left(v_{w_O} \bullet v_{w_I}\right)}{\sum_{w=1}^{V} exp\left(v_w \bullet v_{w_I}\right)}. \tag{3}$$

Where w_O is the context word, w_I is the target word, v_{w_O} and v_{w_I} are the word vector representations of the words w_O and w_I, respectively, V is the size of the vocabulary list, and " $\bullet$ " denotes the vector dot product.

The model updates the word vectors during training by maximising the above conditional probabilities. The specific steps are as follows:

1. Randomly initialise word vectors for all words.
2. For each word w_t in the corpus, take a context word whose surrounding window size is c.
3. Use the softmax function to calculate $P\left(w_{t+j} | w_t\right)$.
4. Maximise the conditional probability by gradient descent and update the word vectors v_{w_O} and v_{w_O}.

Due to the computational intensity of the softmax function, practical applications often resort to methods like Hierarchical Softmax and Negative Sampling to expedite training processes. In this paper, Hierarchical Softmax is employed.

Hierarchical Softmax leverages a Huffman Tree to mitigate computational complexity. This approach reduces the softmax function, which is the computational burden from $O(V)$ to $O\left(\log V\right)$, significantly enhancing efficiency in processing large vocabulary.

TF-IDF. TF-IDF is a widely utilized technique in text analysis and information retrieval for assessing the significance of words within a document. It computes the product of term frequency (TF) and inverse document frequency (IDF), thereby emphasizing words that are particularly distinctive to the document's content.

The primary role of TF-IDF:

1. TF-IDF effectively emphasises crucial words that distinguish documents by diminishing the importance of common words and amplifying the significance of words appearing in a limited number of documents.
2. TF-IDF serves as a valuable text feature extraction method in tasks such as text classification, clustering, and topic modelling. It transforms textual data into a vector representation conducive to subsequent machine learning processing.
3. In search engines, TF-IDF plays a crucial role in measuring the relevance of document queries. It aids in identifying and returning the most pertinent documents by assessing the importance of terms within documents relative to the entire corpus.

The TF-IDF value is obtained by calculating the product of word frequency (TF) and inverse document frequency (IDF). The specific steps are as follows:

1. Calculate word frequency (TF) Word frequency indicates how often a word appears in a document. Word frequency can be calculated in a number of ways; the most common way is to count the number of times a word appears in a document and normalise it:

   $$TF\left(t,d\right) = \frac{f_{t,d}}{N_d} \, . \tag{4}$$

   Where $f_{t,d}$ denotes the number of occurrences of word t in document d, and N_d denotes the total number of words in document d.
2. Calculating the Inverse Document Frequency (IDF) The inverse document frequency is used to reduce the weight of words that occur commonly in multiple documents and is defined as follows:

   $$IDF\left(t,D\right) = \log\left(\frac{N}{\left|\{d \in D : t \in d\}\right|}\right) . \tag{5}$$

   where N denotes the total number of documents and $\{d\epsilon D : t\epsilon d\}$ denotes the number of documents containing the word t.
3. Calculation of TF-IDF values Finally, the TF-IDF value is obtained by multiplying TF and IDF:

   $$TF\text{-}IDF\left(t,d,D\right) = TF\left(t,d\right) \times IDF\left(t,D\right) \; . \tag{6}$$

Web Feature Vectorisation. This paper compiles a comprehensive dataset of POST messages from gambling websites, referred to as the Gambling Website POST Message Site Merge Dataset (GPMD). This dataset comprises three components: a list of URL words, a list of cookie parameter words in the request header, and a list of parameter words in the request body.

Firstly, the list of words in the URL of each website in the dataset, the list of words in the Cookie parameter in the request header, and the list of words in the request body parameter are extracted, and the three-word lists are integrated. The integration is the union of sets to ensure that all relevant words from each website are included in the final complete word list. These lists are then combined to calculate the global TF-IDF weights. Analyzing the website data reveals that gambling websites generated by the same template have nearly identical cookie parameters, and the cookie parameters for different behaviours on the same website are also broadly consistent. Therefore, the weight of the cookie parameters is increased when training the Word2Vec model.

The input data consists of a list of URL words for each website, cookie parameter words in the request header, and parameter words in the request body. The structure of this data is shown in Table 1 below.

The data from the above table is used as input. The data is subdivided, the lists are merged, and the TF-IDF weights are calculated. The Word2Vec model is trained, and the word list of each POST message is converted into a TF-IDF-weighted average word vector, which is then saved as a CSV file. The structure of the output data is shown in Table 2.

Table 1. Input data

WebName	RequestLineKey	BodyParam	CookieParam
Name1	['word1','word2', ···]	['param1','param2', ···]	['cookie1','cookie2', ···]
Name2	['word3','word4', ···]	['param3','param4', ···]	['cookie3','cookie4', ···]
...	...	...	...

Table 2. Output data

Dim_0	Dim_1	Dim_2	···	Dim_499	WebName
0.0750426	0.1420163	0.2485222	···	-0.1477123	Name1
0.0622372	0.1342609	0.1215802	···	-0.0396667	Name2
...	...	...	...	...	...

3.3. Web Site Clustering

Hierarchical Clustering. Hierarchical clustering is a powerful method for uncovering hierarchical structures and intrinsic relationships in data. It effectively groups similar data points by calculating a distance matrix, gradually merging the most similar clusters, and updating the distance matrix. In practice, hierarchical clustering is widely used in data analysis, grouping, pattern recognition, and simplification.

There are two main types of hierarchical clustering:

1. Agglomerative Hierarchical Clustering (AHC) is a bottom-up approach where each data point starts as its cluster. The most similar clusters are progressively merged until all data points are combined into a single cluster or a predetermined number of clusters is reached.
2. Divisive Hierarchical Clustering (DHC) is a top-down approach where all data points initially form one cluster. This cluster is progressively split into smaller clusters until each data point stands alone as its cluster or a predefined number of clusters is reached.

The following are the detailed implementation steps for cohesive hierarchical clustering:

1. Calculate the distance matrix: Calculate the distance or similarity among all data points. Commonly used distance metrics include Euclidean distance, Manhattan distance and cosine similarity.

$$d_{ij} = \sqrt{\sum_{k=1}^{n} \left(x_{ik} - x_{jk}\right)^2} \, . \tag{7}$$

   Where d_{ij} denotes the Euclidean distance between data points i and j, and x_{ik} and x_{jk} denote the values of i and j in the kth dimension, respectively.
2. Merge the most similar clusters: At each step, find the two closest clusters and merge them into a new cluster.

$$d\left(A \bigcup B, C\right) = \min \left\{d\left(a, c\right) : a \in A, c \in C\right\} . \tag{8}$$

Where A and B are the two clusters to be merged, and $d\left(A \cup B, C\right)$ is the distance between the new cluster and the other clusters.

3. Update Distance Matrix: Update the distance matrix to reflect the distance between the new cluster and all other clusters.
   a. Single Linkage method: The distance between a new cluster and other clusters is the smallest distance between any two points in its constituent clusters.
   $$d\left(A \cup B, C\right) = \min\left\{d\left(a, c\right) : a \in A, c \in C\right\}. \tag{9}$$
   b. Complete Linkage: The distance of a new cluster from other clusters is the maximum distance between any two points in its constituent clusters.
   $$d\left(A \cup B, C\right) = \max\left\{d\left(a, c\right) : a \in A, c \in C\right\}. \tag{10}$$
   c. Average Linkage: The distance between a new cluster and other clusters is the average of the distances between all pairs of points in its constituent clusters.
   $$d\left(A \cup B, C\right) = \frac{1}{|A||C|} \sum_{a \in A, c \in C} d\left(a, c\right). \tag{11}$$
   d. Ward's Method: Ward's method defines the distance between clusters by minimising the Sum of Squared Errors (SSE) within the merged cluster.
   $$\triangle SSE = \frac{|A||B|}{|A| + |B|} \left\|\overline{x}_A - \overline{x}_B\right\|^2 . \tag{12}$$
   Where $|A|$ and $|B|$ are the number of data points in clusters A and B, and $\overline{x}_A$ and $\overline{x}_B$ are the centres of mass of clusters A and B. Ward's method tends to generate clusters that are similar in size and regular in shape
4. Repeat steps 2 and 3 until all data points are combined into a single cluster or a predetermined number of clusters is reached.

Clustered Website Templates. In this paper, the text data is represented as high-dimensional vectors using the Word2Vec model, which effectively captures the data's structure. By applying TF-IDF weighting, the importance of keywords is enhanced, reducing the influence of common but non-discriminatory words. This approach improves the accuracy of clustering the websites.

Because this paper cannot accurately distinguish which class of template a gambling website belongs to based on its visible form, there are no template-based labels in the Gambling Site POST Message Merge Dataset (GPMD). Hierarchical clustering, particularly agglomerative hierarchical clustering, does not require a predetermined number of clusters. Instead, clustering is controlled through a distance threshold, allowing the identification of natural clusters by analyzing the structure of the data at different scales.

Using the Ward linkage method minimizes the variance of the clusters at each merge. This approach ensures that each merge aims to preserve the tightness within the cluster, thereby promoting the formation of more compact clusters.

Calculating the distance matrix using Euclidean distance ensures that the distance metric in the clustering process is standardized and valid in high-dimensional space. The following are the specific data processing steps.

Firstly, the preprocessed data was loaded, where each text was converted into a high-dimensional vector, as shown in the table. We extracted 500 columns as features to capture the primary information of the text. These high-dimensional vectors reflect the structure of the data well, and by applying TF-IDF weighting, we further enhance sensitivity to critical words. This weighting method helps mitigate the impact of frequently occurring but non-discriminative words, thereby improving the accuracy of clustering.

In the clustering process, the distance matrix of the data was first calculated using Euclidean distance as the metric. This ensures that the distance metric used in the clustering process is standardized and valid in high-dimensional spaces. Hierarchical clustering through the Ward linkage method allows us to minimize the intra-cluster variance at each merge, thereby maintaining the tightness of the clusters. After clustering is complete, we incorporate the clustering results into the raw data and output the WebName tags in each class. The format is shown in Table 3. This helps us understand the composition of each cluster and its characteristics. To better understand the clustering results, we further reduce the dimensionality of the high-dimensional data to 2D using the t-SNE (t-Distributed Stochastic Neighbor Embedding) method and plot a scatter plot. t-SNE effectively preserves the local structure of high-dimensional data, enabling us to visualize the distribution and relationships among different clusters. In the two-dimensional scatterplot generated after t-SNE reduction, each point is colour-coded according to the cluster to which it belongs. This allows clear visualization of the internal structure and relationships between clusters.

Table 3. Web site clustering

Cluster	Website Name
Cluster 0	‘WebName1’ ‘WebName2’ $\cdots$
Cluster 2	‘WebName3’ ‘WebName4’ $\cdots$
...	...
Cluster N	‘WebName 174’ ‘WebName175’ $\cdots$

3.4. Abnormal Website POST Classification

Mutual Information Act. The mutual information method is a statistical measure used to quantify the amount of information one random variable provides about another. It assesses the dependence or correlation between two variables, making it valuable in various fields such as feature selection, feature engineering, image processing, and information retrieval.

In machine learning, the mutual information method is crucial in evaluating and selecting features that strongly correlate with the target variable. By identifying such features, models can achieve improved performance and generalization capabilities. In image registration, mutual information can assess the similarity between two images, aiding their alignment by quantifying how much information is shared. Moreover, the mutual information method helps evaluate the relevance between documents and queries in information retrieval tasks. This evaluation enhances the accuracy of retrieval results by identifying documents that contain information most closely related to the user’s query.

Mutual information quantifies the interdependence between two random variables X and Y, defined as:

$$I\left(X;Y\right)=\sum_{y\in Y}\sum_{x\in X}p\left(x,y\right)\log\left(\frac{p\left(x,y\right)}{p(x)p(y)}\right). \tag{13}$$

Where $p\left(x,y\right)$ is the joint probability distribution of X and Y, and $p(x)$ and $p(y)$ are the marginal probability distributions of X and Y, respectively.

Particle Swarm Algorithm(PSO). Particle Swarm Algorithm (PSO) is an optimisation algorithm based on group intelligence and is mainly used to solve complex optimisation problems. It is inspired by group behaviours such as flocks of birds foraging for food and schools of fish swimming.PSO finds the optimal solution to a problem by simulating individuals (particles) moving and collaborating in the search space.

PSO is widely used in machine learning to optimise model parameters, function optimisation, path planning, engineering optimisation, etc. The PSO algorithm searches for the optimal solution by updating the velocity and position of each particle. Each particle is represented in the search space as a position vector x_i and a velocity vector v_i. The particle adjusts its velocity and position at each step based on its own experience and the experience of its neighbours. Following is the procedure of the particle swarm algorithm PSO:

1. Initialisation
   Initialise the position and velocity of the particles and set the initial parameters such as the number of particles, maximum number of iterations, individual learning factor and social learning factor.

$$x_i(0)\sim U\left(x_{min},x_{max}\right). \tag{14}$$

$$v_i(0)=U\left(v_{min},v_{max}\right). \tag{15}$$

   Where U denotes a uniform distribution, x_{min} and x_{max} are position ranges, and v_{min} and v_{max} are ranges of velocities.
2. Speed Update
   The velocity of each particle is updated according to the following equation:

$$v_i\left(t+1\right)=w\times v_i(t)+c_1\times r_1\times\left(p_i-x_i(t)\right)+c_2\times r_2\times\left(g-x_i(t)\right). \tag{16}$$

   w is the inertia weight, which controls the effect of the particle's previous velocity; c_1 and c_2 are the individual learning factor and the social learning factor, respectively, which control the extent to which the particle moves towards its optimal position and the global optimal position. r_1 and r_2 are random numbers between [0,1]. p_i is the historical optimal position of the particle i. g is the global optimal position found amongst all particles.
3. Location Updates
   The position of the particle is updated according to the following equation:

$$x_i\left(t+1\right)=x_i(t)+v_i\left(t+1\right). \tag{17}$$

4. Evaluating and updating optimal solutions
   Each particle's fitness is evaluated at each step, and the particle's historical best position p_i and global best position g:

$$if\ \ f\left(x_i\left(t+1\right)\right) < f\left(p_i\right)\ \ then\ \ p_i = x_i\left(t+1\right), if\ \ f\left(p_i\right) < f(g)\ \ then\ \ g = p_i\,. \tag{18}$$

Support Vector Machine (SVM). Support Vector Machine (SVM) is a supervised learning algorithm widely used in classification and regression problems. The basic idea of SVM is to find an optimal hyperplane to maximise the classification interval (Margin) to achieve good classification performance.

SVMs were initially designed for binary classification problems but can be extended to multiclassification problems using various strategies. Common approaches are One-vs-Rest (OvR) and One-vs-One (OvO).

1. One-vs-Rest(OvR):
   The OvR method bisects each category with other categories to build K classifiers (for a K-class problem)—for each classifier.

$$f_k(x) = sign\left(w_k \times x + b_k\right)\ . \tag{19}$$

2. One-vs-One (OvO):
   The OvO method builds one classifier for every two categories, for a total of $\frac{K(K-1)}{2}$ classifiers. Each classifier $f_{i,j}(x)$ distinguishes category i from category j.

$$f_{ij}(x) = sign\left(w_{i,j} \times x + b_{i,j}\right)\ . \tag{20}$$

   The final decision is determined through a voting mechanism where each classifier votes for its category, and the category with the most votes is the final classification result:

$$Class(x) = \arg\max_k \sum_{i,j} \delta\left(f_{i,j}(x) = k\right). \tag{21}$$

Support Vector Machines are powerful classification algorithms particularly suitable for high-dimensional data. By introducing kernel functions, SVMs can handle linearly indivisible problems.

Single POST Classification. Based on the cluster analysis of gambling websites, this paper assigns labels to each POST request in the single POST dataset (GSPD) of gambling websites. Analysis reveals that cookie parameters in the request headers of different behaviours within the same gambling website are broadly consistent, and those of websites sharing the same template exhibit high similarity. Therefore, using the TF-IDF method, this study utilizes the request line, cookie parameters from the request header, and request body parameters as input for textual feature extraction. These components collectively form a comprehensive feature vector. Subsequently, the mutual information method is employed for feature selection, retaining features with mutual information scores greater than 0.

After the feature selection is completed, in order to optimise the parameters c and gamma of the Support Vector Machine (SVM) model, this paper adopts the Particle Swarm Optimisation (PSO) algorithm to perform parameter tuning by minimising the negative accuracy on the training data. The specific processing steps are as follows:

1. It reads data from a single POST dataset and extracts features and labels. The dataset contains request lines, request body parameters, cookie parameters, and labels. To evaluate the model performance, the dataset is divided into a training set and a test set, with 80% and 20%, respectively. The format of the dataset is shown in Table 4.

Table 4. POST dataset format

POST Name	RequestLine	CookieParam	BodyParam	Flag
Name1	HTTP Method URL1 HTTP Version	[‘cookie1’,· · ·]	[‘param1’,· · ·]	1
Name2	HTTP Method URL2 HTTP Version	[‘cookie2’,· · ·]	[‘param2’,· · ·]	2
. . .	. . .	. . .	. . .	. . .
NameN	HTTP Method URLN HTTP Version	[‘cookieM’,· · ·]	[‘paramM’,· · ·]	m

   In Table 4, RequestLine denotes the request line, CookieParam denotes the cookie parameter in the request header, and BodyParam denotes the request body parameter.CookieParam and BodyParam contain word lists, and Flag is the class label labelled according to the website clustering. In addition to the above four parts, the POST dataset also includes the behaviour of the POST message, the name of the website to which it belongs, the Cookie, the request body, etc. Table 4 only lists the content required for classification in this paper.
2. Textual features from request lines, request body parameters and cookie parameters are extracted using the TF-IDF method. A FeatureUnion is constructed to facilitate processing where TF-IDF transformation is separately applied to each component. Subsequently, the mutual information method is employed for feature selection, retaining features with mutual information scores greater than 0. This approach aims to reduce feature redundancy and enhance the model’s overall performance.
3. Parameter tuning is conducted using the Particle Swarm Optimization (PSO) method to find the optimal SVM parameters. The objective function used is the negative accuracy of the SVM model on the training set. The PSO algorithm simulates the flight of particles in the solution space to search for the best combination of SVM parameters, aiming to maximize the model’s accuracy on the training data.
4. The SVM model is trained using the optimised parameters, and predictions are made using a divided test set to evaluate the model’s performance.

Through the above steps, this paper effectively classifies POST requests for gambling websites, confirming the significance of the clustering effect and demonstrating high model performance.

4. Experiments and Analysis

In this section, experiments are conducted to evaluate the performance of the proposed method for identifying anomalous POST traffic based on website templates. The experiments were performed on a server with an AMD EPYC 7T83 64-core processor, 90 GB of RAM, and an NVIDIA RTX 4090 GPU with 24 GB of video memory. The system has a 30 GB capacity for the system disk and 50 GB for the data disk. It runs Ubuntu 20.04 with Python 3.8 installed, PyTorch 2.0.0, and CUDA 11.8 support.

Dataset: This paper uses Fiddler, a web debugging proxy tool, to capture POST messages about gambling websites' logins, registrations, bank card additions, betting, and other behaviours. Since some POST messages are repeated, this paper removes duplicate and noisy data. It collates each POST message into tabular data by removing the request line, cookies in the request header, and the empty request body. Data, and finally, 6525 POST data and 175 gambling websites are collated. Then, the complete URL is extracted from the request line, and the words after the URL domain name are extracted for saving, the cookie parameters are extracted and saved, and the parameters in the request body are extracted and saved.

Evaluation metrics: In this paper, four evaluation metrics, accuracy, precision, recall, and F1-score, are used, and the confusion matrix for the experiments in this paper is shown in Table 5. The confusion matrix consists of four values: true positive (*TP*), true negative (*TN*), false positive (*FP*), and false negative (*FN*). Where *TP* denotes the number of samples that were correctly classified as that class for a given class a, *FP* denotes the number of samples that were incorrectly classified as that class for a given class a (i.e., samples belonging to other classes were predicted to be in that class), *FN* denotes the number of samples that were incorrectly classified as other classes for a given class a (i.e., samples actually belonging to that class were predicted to be in that other class), and *FN* denotes the number of samples that were correctly classified as other (i.e., the number of samples that belonged to other classes and were not predicted to be that class). *TN* denotes the number of samples correctly classified as other for a class a (i.e., the number of samples that belonged to that class and were not predicted to be that class).

Table 5. Output data

	Predicted 0	Predicted 1	Predicted 2	···	Predicted N
Actual 0	*TP*	*FP*	*FP*	···	*FP*
Actual 1	*FN*	*TP*	*FP*	···	*FP*
Actual 2	*FN*	*FN*	*TP*	···	*FP*
···	···	···	···	···	···
Actual N	*FN*	*FN*	*FN*	···	*TP*

Accuracy is the proportion of total samples that the classifier predicts correctly. The formula is:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} . \tag{22}$$

The precision rate is the proportion of all samples predicted to be in the positive category that are actually in the positive category. The formula is:

$$Precision = \frac{TP}{TP + FP} . \tag{23}$$

Recall is the proportion of all samples in the positive category that are correctly predicted to be in the positive category. The formula is:

$$Recall = \frac{TP}{TP + FN} . \tag{24}$$

The F1 score is the reconciled average of precision and recall and is used to assess the precision and recall of the model in a combined manner. The formula is:

$$F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} . \tag{25}$$

The above four evaluation metrics, accuracy, precision, recall, and F1, are important; the closer to 1 the four metrics are, the better the model's performance in this paper.

Web site clustering evaluation. Firstly, this paper calculates the TF-IDF weights for the whole dataset, and the IDF values for each word are stored in a dictionary, which is once again used for subsequent use in generating word vectors.

In this paper, a Word2Vec model is trained on a curated dataset of gambling websites. The word vectors are configured with a dimensionality of 500, and a context window size of 2 is employed. Due to the dataset's preprocessing, which involved removing duplicates, the minimum word frequency threshold is set to 1. The training process iterates over 80 rounds, employing the Skip-gram method with hierarchical softmax for training the Word2Vec model.

This paper utilizes three main components: the URL vocabulary from the dataset, cookie parameters in the request header (given twice the weighting), and parameters from the request body. The vector representation of each sentence synthesized from these components is computed using TF-IDF weights, where the weights of the parameters are also considered in the calculation.

Save the file using the trained Word2Vec model and the generated sentence vectors.

Using the saved vector file mentioned earlier, this paper clusters 175 gambling websites from the dataset using hierarchical clustering. Specifically, Agglomerative Clustering is employed with Ward's method as the linkage criterion. The clustering results are depicted in Fig. 3 (Website Template Clustering).

According to Figure 3, this paper clusters 175 websites into ten classes: Cluster 0 contains 14 websites, Cluster 1 contains 16 websites, Cluster 2 contains 45 websites, Cluster 3 contains 6 websites, Cluster 4 contains 16 websites, Cluster 5 contains 11 websites, Cluster 6 contains 10 websites, Cluster 7 contains 13 websites, Cluster 8 contains 33 websites, and Cluster 9 contains 11 websites. The paper analyzes the data and identifies several clusters that contain similar templates, such as Wanjiao.com, Fujiao.com, Dejiao.com, Xiangjiao.com, Longjiao.com, and various lottery websites, as well as similar sports gambling websites, following the processing described in the paper.

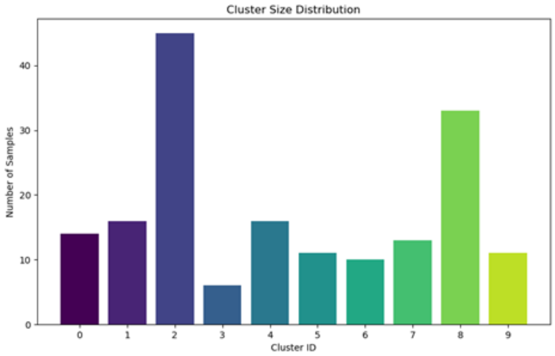

Fig. 3. Cluster Size Distribution

Based on the clusters clustered by the above hierarchical clustering, Fig. 4 (Cluter 3 homepage) and Fig. 5 (Cluter 4 homepage) are screenshots of the websites' home pages under the same class templates selected above.

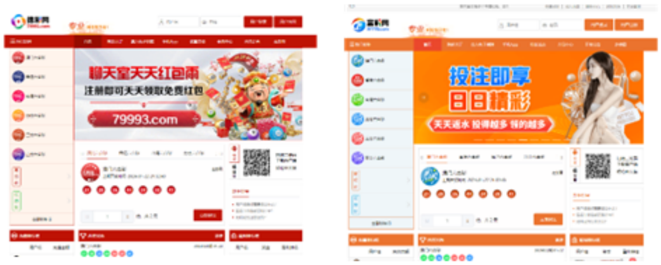

Fig. 4. Cluter 3 homepage

Screenshots of the home pages of two randomly selected websites from Cluster 3 are presented in Figure 4: The first shows the home page of DeColour, and the second displays the home page of FooColour. Similarly, screenshots of two randomly selected websites from Cluster 4 are shown in Figure 5: the first depicts the home page of BoyeSports.com, and the second shows the home page of FourSeasonsSports4.vip.com. By comparing the screenshots in Fig. 4 and Fig. 5, it becomes evident that DeColour and FooColour share the same template, as do BoyeSports and FourSeasonsSports4.vip.com. This illustrates the effectiveness of our clustering approach in correctly grouping websites that utilize identical templates.

The 500-dimensional data was downscaled to 2 dimensions using t-SNE to visualise the effect of the above clustering of gambling sites, as shown in Fig. 6 (Clustered scatter-plot).

Fig. 5. Cluter 4 homepage

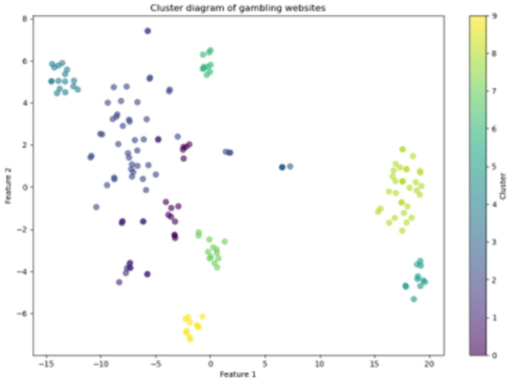

Fig. 6. Cluster diagram of gambling websites

In the t-SNE scatterplot, each point's position reflects its relationships within the 500-dimensional feature space. Different colours in Figure 6 represent distinct clustering results. The plot reveals that a few points are situated near the boundaries of multiple clusters, possibly indicating feature ambiguity or noise that complicates clear distinction after dimensionality reduction. However, multiple clearly defined clusters in Figure 6 indicate significant feature similarity among points in the original high-dimensional space, underscoring the effectiveness of the clustering.

Evaluation of POST traffic classification. Based on the websites' clustering results, this paper annotates each POST in the crawled dataset with a corresponding cluster label. The dataset comprises 6525 POST entries collated from the websites.

This paper applies an 80%-20% split to the dataset, with 80% used for training and 20% for testing. The dataset includes three components: the request line, cookie parameters in the request header, and parameters from the request body, which are selected as input data. Additionally, three feature extraction pipelines are employed to process each feature column and transform the text into TF-IDF feature vectors. Subsequently, mutual information between each feature and the label is computed, and features with mutual information scores greater than 0 are selected for further analysis.

Define the objective function of PSO optimisation to assess the parameter combination's goodness using the SVM model's accuracy. PSO defines the support vector machine model 'c' parameter range to be [0.01,10]. The parameter range of 'gamma' is [0.0001,1]; by PSO optimising the objective function, the best parameters are obtained as 4.2414 and 0.0790, respectively. Using the best parameters to train the SVM model, the test set is predicted, and the classification test results are shown in Table 6.

Table 6. Message classification results

Cluter	Precision	Recall	F1 Score	Actual sample size
0	1.0000	1.0000	1.0000	211
1	1.0000	1.0000	1.0000	397
2	1.0000	1.0000	1.0000	34
3	0.9820	1.0000	0.9909	109
4	1.0000	1.0000	1.0000	230
5	1.0000	1.0000	1.0000	7
6	1.0000	1.0000	1.0000	51
7	1.0000	1.0000	1.0000	60
8	1.0000	0.9887	0.9943	177
9	1.0000	1.0000	1.0000	29
Accuracy			0.9985	1305
Macro avg	0.9982	0.9989	0.9985	1305
Weighted avg	0.9985	0.9985	0.9985	1305

As shown in Table 6, the overall accuracy (Accuracy) is 0.9985, indicating that only a few of the 1305 test samples were misclassified. The proportion of all samples predicted to be in the positive class that was actually in the positive class, i.e., the precision rate, is almost always 1, indicating that the model predicts very accurately. The proportion of all samples that were positively classified and correctly predicted to be positive, i.e., the recall, was also almost always 1, indicating that the model identified positively classified samples very well. The harmonic mean of precision and recall, or F1 score, is also close to 1, indicating that the model performs very well in all categories.

Table 7 shows the results after applying the dataset's Random Forest, K-Nearest Neighbors (KNN), Naive Bayes, XGBoost and NODE (Neural Oblivious Decision Ensembles) deep learning models. Results of the classification performed.

Table 7. Classification results by model

Model	Accuracy	Precision	Recall	F1-Score
SVC	0.9985	0.9985	0.9985	0.9985
Random Forest	0.9900	0.9900	0.9900	0.9897
KNN	0.9785	0.9792	0.9785	0.9784
Naive Bayes	0.9456	0.9429	0.9456	0.9415
XGBoost	0.9885	0.9886	0.9885	0.9883
NODE	0.9862	0.9900	0.9900	0.9900

As depicted in Table 7, aggregating all metrics, the Support Vector Machine (SVC) with optimized parameters demonstrates superior performance in this classification task, achieving the highest accuracy, precision, recall, and F1 score. The experimental outcomes underscore its efficiency in processing and classifying data, effectively adapting to accurately categorizing POST message traffic from individual gambling websites based on features such as the request line, request header cookie parameters, and request body parameters.

5. Conclusions and Outlook

This paper proposes and validates a method for classifying abnormal POST traffic based on website templates. Our experiments demonstrate the method's effectiveness and high performance in identifying and classifying abnormal traffic on gambling websites. Firstly, using the Fiddler tool, we capture POST messages related to login, registration, adding bank cards, betting, and other behaviours on gambling websites. After removing duplicates and noisy data, we compile 6525 POST entries. We perform feature extraction on URLs, cookie parameters in the request header, and parameters in the request body using TF-IDF and Word2Vec, assigning appropriate weights to different features for efficient representation. Secondly, we use Agglomerative Clustering with Ward's method to cluster 175 gambling websites. The results show that websites with similar templates and structures are successfully grouped, verifying the effectiveness of our method in recognizing website templates. Based on these clustering results, we label the POST data and optimize the classification model's parameters using a Support Vector Machine (SVM) combined with Particle Swarm Optimization (PSO). The experimental results indicate that the classification accuracy on the test set is 0.9985, with precision, recall, and F1-score all close to 1, demonstrating the model's excellent performance across all categories. Finally, we experimentally verify that the SVM model is better suited to classify the traffic of individual gambling website POST messages based on the request line, request header cookie parameters, and request body parameters compared to other models. This highlights the method's high practical value in effectively identifying and classifying anomalous network traffic. In summary, the method presented in this paper shows excellent performance and application potential in classifying gambling website POST traffic. Future research will optimize feature extraction and model training methods and extend their application to other web traffic classification tasks.

In this paper, we have collected data from various existing gambling websites, covering all known types of gambling website templates. However, to address the emergence of new gambling website templates in the future, we propose further investigation into recognition methods using unsupervised learning techniques to tackle these new challenges.

Acknowledgments. This work was Supported by Key Lab of Information Network Security, Ministry of Public Security,the National Natural Science Foundation of China under Grants 61672338, the Natural Science Foundation of Shanghai under Grant 21ZR1426500.

References

1. Auer, M., Griffiths, M.D.: Using artificial intelligence algorithms to predict self-reported problem gambling with account-based player data in an online casino setting. Journal of Gambling Studies 39(3), 1273–1294 (2023)
2. Cai, S., Han, D., Li, D.: A feedback semi-supervised learning with meta-gradient for intrusion detection. IEEE Systems Journal 17(1), 1158–1169 (2022)
3. Cernica, I., Popescu, N.: Computer vision based framework for detecting phishing webpages. In: 2020 19th RoEduNet Conference: Networking in Education and Research (RoEduNet). pp. 1–4. IEEE (2020)
4. Chen, Y., Zheng, R., Zhou, A., Liao, S., Liu, L.: Automatic detection of pornographic and gambling websites based on visual and textual content using a decision mechanism. Sensors 20(14), 3989 (2020)
5. Fan, Y., Yang, T., Wang, Y., Jiang, G.: Illegal website identification method based on url feature detection. Comput. Eng 44, 171–177 (2018)
6. Gu, Z., Gou, G., Liu, C., Yang, C., Zhang, X., Li, Z., Xiong, G.: Let gambling hide nowhere: Detecting illegal mobile gambling apps via heterogeneous graph-based encrypted traffic analysis. Computer Networks 243, 110278 (2024)
7. Gupta, J., Pathak, S., Kumar, G.: Aquila coyote-tuned deep convolutional neural network for the classification of bare skinned images in websites. International Journal of Machine Learning and Cybernetics 13(10), 3239–3254 (2022)
8. Jain, A.K., Gupta, B.B.: A machine learning based approach for phishing detection using hyperlinks information. Journal of Ambient Intelligence and Humanized Computing 10, 2015–2028 (2019)
9. Kairouz, S., Costes, J.M., Murch, W.S., Doray-Demers, P., Carrier, C., Eroukmanoff, V.: Enabling new strategies to prevent problematic online gambling: A machine learning approach for identifying at-risk online gamblers in france. International Gambling Studies 23(3), 471–490 (2023)
10. Li, D., Han, D., Weng, T.H., Zheng, Z., Li, H., Liu, H., Castiglione, A., Li, K.C.: Blockchain for federated learning toward secure distributed machine learning systems: a systemic survey. Soft Computing 26(9), 4423–4440 (2022)

11. Li, D., Han, D., Zheng, Z., Weng, T.H., Li, K.C., Li, M., Cai, S.: Does short-and-distort scheme really exist? a bitcoin futures audit scheme through birch & bpnn approach. Computational Economics 63(4), 1649–1671 (2024)
12. Li, J., Han, D., Weng, T.H., Wu, H., Li, K.C., Castiglione, A.: A secure data storage and sharing scheme for port supply chain based on blockchain and dynamic searchable encryption. Computer Standards & Interfaces 91, 103887 (2025)
13. Li, J., Han, D., Wu, Z., Wang, J., Li, K.C., Castiglione, A.: A novel system for medical equipment supply chain traceability based on alliance chain and attribute and role access control. Future Generation Computer Systems 142, 195–211 (2023)
14. Li, L., Gou, G., Xiong, G., Cao, Z., Li, Z.: Identifying gambling and porn websites with image recognition. In: Advances in Multimedia Information Processing–PCM 2017: 18th Pacific-Rim Conference on Multimedia, Harbin, China, September 28-29, 2017, Revised Selected Papers, Part II 18. pp. 488–497. Springer (2018)
15. Liu, D., Lee, J.H., Wang, W., Wang, Y.: Malicious websites detection via cnn based screenshot recognition. In: 2019 International Conference on Intelligent Computing and its Emerging Applications (ICEA). pp. 115–119. IEEE (2019)
16. Sun, G., Ye, F., Chai, T., Zhang, Z., Tong, X., Prasad, S.: Gambling domain name recognition via certificate and textual analysis. The Computer Journal 66(8), 1829–1839 (2023)
17. Syahputra, H., Wibowo, A.: Comparison of support vector machine (svm) and random forest algorithm for detection of negative content on websites. Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI) 9(1), 165–173 (2023)
18. Urvoy, T., Chauveau, E., Filoche, P., Lavergne, T.: Tracking web spam with html style similarities. ACM Transactions on the Web (TWEB) 2(1), 1–28 (2008)
19. Wang, C., Xue, P., Zhang, M., Hu, M.: Identifying gambling websites with co-training. In: SEKE. pp. 598–603 (2022)
20. Wang, C., Zhang, M., Shi, F., Xue, P., Li, Y.: A hybrid multimodal data fusion-based method for identifying gambling websites. Electronics 11(16), 2489 (2022)
21. Yang, P., Zhao, G., Zeng, P.: Phishing website detection based on multidimensional features driven by deep learning. IEEE access 7, 15196–15209 (2019)
22. Zhang, W., Jiang, Q., Chen, L., Li, C.: Two-stage elm for phishing web pages detection using hybrid features. World Wide Web 20, 797–813 (2017)
23. Zhao, J., Shao, M., Peng, H., Wang, H., Li, B., Liu, X.: Porn2vec: A robust framework for detecting pornographic websites based on contrastive learning. Knowledge-Based Systems 228, 107296 (2021)
24. Zhou, S., Ruan, L., Xu, Q., Chen, M.: Multimodal fraudulent website identification method based on heterogeneous model ensemble. China Communications 20(5), 263–274 (2023)
25. Zhou, S., Xu, C., Xu, R., Ding, W., Chen, C., Xu, X.: Image recognition model of fraudulent websites based on image leader decision and inception-v3 transfer learning. China Communications 21(1), 215–227 (2024)
26. Zuhair, H., Selamat, A.: Phishing hybrid feature-based classifier by using recursive features subset selection and machine learning algorithms. In: Recent Trends in Data Science and Soft Computing: Proceedings of the 3rd International Conference of Reliable Information and Communication Technology (IRICT 2018). pp. 267–277. Springer (2019)

Zhimin Feng is currently pursuing the M.S.degree with the School of Information Engineering, Shanghai Maritime University, Pudong, China. His current research interest is network security.

Dezhi Han received the B.S. degree in applied physics from the Hefei University of Technology, Hefei, China, in 1990, and the M.S. and Ph.D. degrees in computing science from the Huazhong University of Science and Technology, Wuhan, China, in 2001 and 2005, respectively. He is currently a Professor with the Department of Computer, Shanghai Maritime University, Pudong, China, in 2010. His current research interests include cloud and outsourcing security, blockchain, wireless communication security, network, and information security.

Songyang Wu is a researcher and director of the Cyber Security Center of the Third Research Institute of the Ministry of Public Security (MPS) and also serves as the deputy director of the National Engineering Research Center for Network Security Level Protection and Security Technologies and the executive deputy director of the Key Laboratory of the Ministry of Public Security of the Ministry of Information Network Security, etc. He received his B.S. degree in Computer Science and Technology from Tongji University in 2005 and his Ph.D. in Computer Application from Tongji University in 2011. He joined the Center of the Ministry of Public Security in the same year. He received his PhD in Computer Application from Tongji University in 2011. He joined the Network Security Center of the Third Research Institute of the Ministry of Public Security in the same year. His research interests include cybercrime investigation, electronic data forensics, ample data security, and artificial intelligence security.

Wenqi Sun, Associate Researcher and Research Engineer at the Cyber Security Center of the Third Research Institute of the Ministry of Public Security; she received her B.S. degree in Computer Science and Technology from Northeastern University in 2010 and her Ph.D. degree in Computer Science and Technology from Tsinghua University in 2016. She joined the Cyber Security Center of the Third Research Institute of the Ministry of Public Security in 2018; her current research interest is in cybercrime investigation.

Shuxin Shi received an M.S. in Computer Science and Technology from Shanghai Maritime University, Pudong, China, in 2024 and is currently pursuing a Ph.D. in the School of Information Engineering. His current research interest is network security.

Received: July 28, 2024; Accepted: September 17, 2024.