Identifying Incoherent Search Sessions: Search Click Fraud Remediation Under Real-World Constraints

Runze Zhang[1], Ranjita Pai Sridhar[2], Mingxuan Yao[1], Zheng Yang[1], David Oygenblik[1], Haichuan Xu[1], Vacha Dave[2], Cormac Herley[2], Paul England[2*], Brendan Saltaformaggio[1]
[1]*Georgia Institute of Technology* [2]*Microsoft Corporation*

***Abstract*—Search engines and advertisers continuously suffer substantial financial losses from click fraud, which poses challenges to existing detection algorithms. Even more concerning, despite ongoing advancements, our understanding of click fraud remains limited, leaving room for sophisticated fraudulent techniques to bypass existing detection measures. In this study, we pivot from examining individual search requests to analyzing search sessions, defined as sequences of consecutive search queries made by the same user. We found that benign users exhibit coherent behavior patterns within these sessions, which contrast clearly with those of fraudulent actors. Specifically, legitimate users tend to conduct searches focused on a single topic at a time. In contrast, fraudsters or automated bots often exhibit diverse, illogical, and incoherent search behaviors within a session. To address this behavioral distinction, we propose CoSeC, a system designed to quantify the "incoherence index" of search sessions. CoSeC integrates literal semantic, temporal, and ad-click behavioral features to evaluate sessions' coherence quantitatively. Our evaluation of CoSeC demonstrates high efficacy, achieving a precision of 95.79% and a recall of 92.40% in identifying incoherent sessions, highlighting CoSeC 's substantial potential to enhance real-world click fraud detection.**

1. Introduction

Search engines primarily monetize their platforms through advertising, which constitutes over 40% of the digital advertising market share in the U.S. [1]. These platforms display ads customized to user queries and charge advertisers based on clicks or impressions. However, advertisers increasingly encounter fraudulent ad-click traffic, risking the depletion of their ad budgets. Reports indicate that click fraud caused $84 billion in losses for online advertisers in 2023, with approximately 14% of search ad engagements driven by fraudsters [2], [3]. Without effective methods to identify and prevent payments for fraudulent traffic, search engines risk losing customers due to unsatisfactory advertising outcomes.

Porting traditional click fraud detection methods to search engines presents unresolved research challenges [4]. For instance, ViceROI [5] leveraged aggregated statistical features from scaled search engine traffic to pinpoint websites collaborating with fraudsters for profit through fraudulent ad clicks. Unfortunately, as acknowledged in [5], [6], labeling traffic at a website level is course-grained, leading to false negatives (fraudulent websites frequented by benign users) and false positives (benign websites frequented by both fraudsters and good users). Thus, search engines must employ finer-grained techniques to label individual requests or user profiles.

However, search engines encounter unique challenges in fine-grained fraudulent traffic detection. Ideally, they could leverage existing machine learning algorithms [7]–[11] to detect ad clicks using features from request content (such as IP address, user agent, and device information) or statistical features from consecutive requests (like click frequency and total spanning time). Unfortunately, studies by Zhu et al. [12] and Szurdi et al. [13] revealed that these features are unreliable due to manipulation by fraud campaigns. Mobile in-app ad networks combat manipulated traffic with client-side fingerprint data from enforced SDKs [12], [14]–[17]. However, search engines, which operate primarily within browsers, lack the infrastructure to collect client-side information like user interface layout and bot execution traces. As a result, they have limited visibility and integrity guarantees for gathering confidential features in fine-grained click fraud detection.

This led us to a fundamental question: How can search engines effectively combat click fraud? Our collaborator, Microsoft, provided a unique opportunity to delve into real-world ad-click traffic from a top search engine. By analyzing real-time traffic, we gained insights into fraudsters' strategies for generating fraudulent traffic and evading detection. We noticed that search engines could identify user-level search sessions, i.e., consecutive queries submitted by each unique user's cookie. By examining timestamped consecutive search queries within these sessions, we uncovered an invariant feature that distinguishes organic users from fraudsters. The key lies in coherence: *Humans typically have a single train of thought, and their search behavior tends to produce coherent consecutive queries, while bots generate incoherent query sequences to maximize revenue or engage targeted ads.*

Building upon this key insight, we hypothesize that search engines can improve fraud detection by holistically analyzing the user sessions alongside individual click and pageview

* *Work performed while the author was with Microsoft.*

DOI 10.1109/SP61157.2025.00111

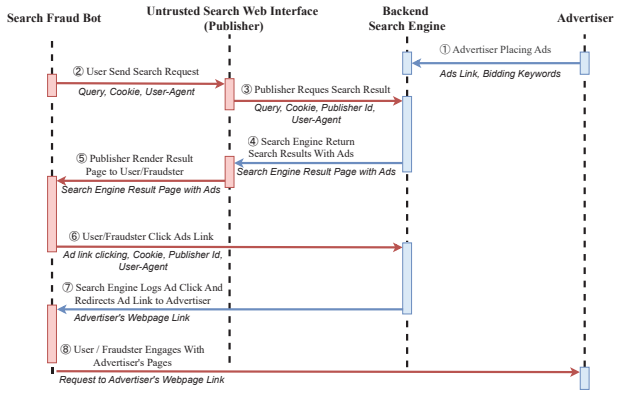

Figure 1: Search Ad-click Fraud Ecosystem. Each arrow represents network requests between entities with transferred information labeled below. Untrusted traffic is in red.

events. To this end, we propose CoSeC[1], an incoherence measurement framework for session-level fraud detection. CoSeC takes the raw search engine traffic as input, identifies search sessions from each user, and extracts literal semantic, temporal, and ad-click behavioral features from any given user's search sessions. It then uses a lightweight sequential classifier to derive a search session's *incoherence index*, which infers whether a search session is fraudulent.

We evaluated CoSeC on our collaborator's data and achieved 95.79% precision and 92.40% recall. Additionally, we show that CoSeC's performance remains robust despite concept drift. It maintains a high accuracy of 92.34% on a different dataset collected one month after the training datasets were collected. Furthermore, we reveal that fraudsters use cutting-edge evasion logic and demonstrate CoSeC to successfully detect this traffic, even though they are actively modifying search queries to build "coherent" sessions. Finally, we made CoSeC's prototype available at https://github.com/CyFI-Lab-Public/COSEC.

2. Motivation

We start with reviewing the background of the search ad ecosystem with a click fraud scenario and fraudsters' motivation in §2.1. We detail the limitations of applying state-of-the-art (SOTA) works to flag search click fraud in §2.2. We present our greatest insight in §2.3 and provide a proof-of-concept measurement using CoSeC in §2.4.

2.1. Click Fraud Ecosystem For Search Engines

Figure 1 presents a simplified model of the search ad's ecosystem, incorporating a typical click fraud scenario. In this model, the *Backend Search Engine* (Column 3 in Figure 1) generates search results and ads from *Advertisers* (Column 4) to the search web interface, which is typically a client-facing publisher. This interface, which may be owned either by the search engine (e.g., *Google.com*, *Bing.com*) or by a third-party sub-syndicator (e.g., Ask.com sub-syndicating Google's results [18]), connects users to search results and ads. In the context of click fraud, an *Untrusted Search Web Interface* (Column 2) represents third-party publishers that may attempt to drive fraudulent ad clicks to increase revenue. Such untrusted publishers may also employ *Search Fraud Bots* (Column 1 in Figure 1) to inflate ad-click volumes. These bots may operate as manipulated botnets [19] and click farms [5], [12]. In Figure 1, we differentiate requests from fraudulent sources (in red) from those originating from legitimate sources (in blue). Pearce et al. [19] provides further insights into fraudsters' motivations.

Initially, advertisers submit bidding keywords to the backend search engine (①), enabling ad targeting aligned with user searches. Subsequently, a fraud bot initiates a search query on the publisher site (②), sending data such as keywords, cookies, and user agents. This request, enriched with telemetry data from the bot, is transmitted to the backend search engine (③). The backend then responds with search results and ads (④ and ⑤), which are displayed to the bot on the publisher's site. The ad URL includes the publisher's ID, linking ad engagements back to the publisher. When the fraud bot clicks an ad (⑥), it sends details such as cookies, user agents, and publisher IDs, enabling the backend to attribute the ad engagement to the publisher and allocate ad revenue accordingly. Upon logging the click, the backend redirects the bot to the advertiser's page (⑦), facilitating a visit to the advertiser's site (⑧). Notably, bots can also interact directly with publisher websites owned by search engines, although these attacks lack a sustainable monetization model, limiting their scalability. This scenario is further discussed in §5.2.

2.2. Limited Fraud Detection By Statistical Method

As highlighted by previous studies [12], [20], fraudsters manipulate traffic and obscure their real IP addresses behind IP proxies, making features from individual search requests unreliable. To derive more robust information from traffic data, several studies [7], [12], [21] leverage *aggregated statistical features of sessions* to enhance fraud detection. These features capture statistical measures across requests within a session, such as click frequency [21] and mobile app active times [12]. In alignment with this approach, we started by examining whether such aggregated features could effectively enable search engines to distinguish fraudulent traffic from organic traffic. For our preliminary analysis, we utilized traffic data provided by our collaborator, collected in February 2024. This dataset includes both fraudulent traffic flagged by our collaborator and organic traffic for comparison. We sampled 10,000 fraudulent sessions and 10,000 organic sessions from this dataset, with sessions identified using user cookies.

Aggregated Statistic Measurement. Our experiment begins with evaluating six statistical metrics relevant to click fraud detection: (1) session length, (2) count of distinct IP addresses per session, (3) average time interval between search requests in a session, (4) standard deviation

1. CoSeC: **Co**herence **Se**quence **C**lassifier

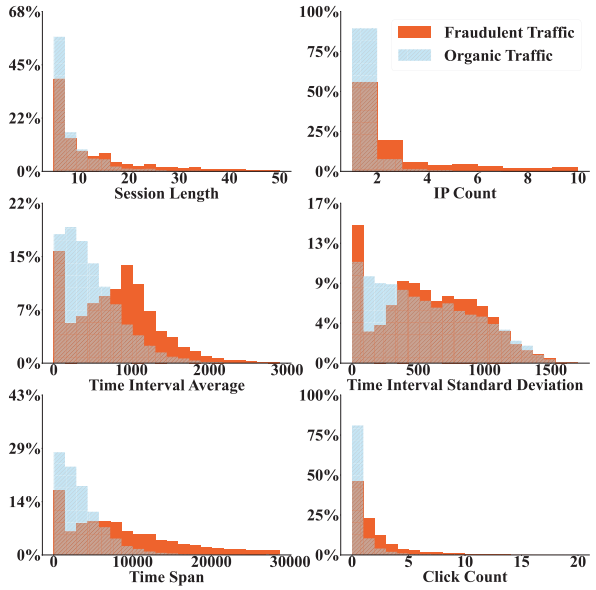

Figure 2: Statistic Measurements Between Organic Sessions and Fraudulent Sessions. The significant overlapping makes fine-grained click fraud detection impossible.

of time intervals between search requests, (5) total session time span, and (6) total click count. These metrics were applied to the dataset. The distributions for these session-based metrics are shown in Figure 2.

The results reveal two key insights. On a positive note, search engines can detect noticeable differences in session-level aggregated features between fraudulent and organic traffic. By applying techniques from ViceROI [5], search engines can aggregate these features at the publisher level and identify unethical publishers as primary sources of fraudulent traffic. However, the significant overlap in each metric between fraudulent and organic traffic, as shown in Figure 2, highlights a challenge. These metrics alone are insufficient for search engines to reliably distinguish fraudulent traffic from organic traffic at a granular level. Consequently, search engines must consider additional methods beyond session-level temporal and ad-click behavior features to improve fraud detection accuracy.

2.3. Key Insight: Session (In)coherence

A distinctive characteristic of the search click fraud scenario caught our attention: the necessity of search queries to generate search result pages and ad links for fraudulent clicking. We hypothesized that these search queries could provide search engines with unique indicators to detect fraudulent traffic. Additionally, as highlighted in [5], [22], fraudsters often generate specific search queries targeting ad clicks associated with higher profitability. To explore this hypothesis, we began our investigation by examining random sessions from the dataset, showcasing examples from both organic and fraudulent traffic in Table 1. Columns 1-3 list the queries, timestamps, and ad counts returned by search engines, respectively.

TABLE 1: SEARCH SESSION EXAMPLES.

Query Keyword	Time Stamp	# Ads
Fraudulent Session Example*, Cosine Similarity Score: 0.089		
top 10 mba online programs†	23:16:32	7
suburbs near shorewood illinois	00:04:05	3
best roblox extension	00:11:18	0
digi canaan	00:16:53	1
brick rigs steam grid	00:30:20	0
best of bowie and queen	00:48:10	0
jessica virginia beach storm	00:55:05	0
you cook by yourself	00:57:08	3
Fraudulent Session Example 1*, Cosine Similarity Score: 0.364		
fractional reserve banking	11:58:12	1
Reserve requirement Wikipedia	11:58:45	0
pet friendly vacations near me	12:02:02	7
pet friendly vacations near me	12:02:04	4
pet friendly vacations near me	12:02:05	4
live edge lumber near me	12:22:01	0
mendels genetics†	12:22:03	10
rolex explorer†	13:13:14	26
Organic Session Example 1*, Cosine Similarity Score: 0.687		
home pack 5kg cement mortar mix	03:37:13	2
home pack 5kg cement mortar mix	03:41:09	3
cement mortar mix	03:55:17	2
dry mortar mix for blocks	04:10:52	3
paint paddle mixer for drill	04:17:30	3
cement paddle mixer for drill	04:24:39	4
Organic Session Example 2*, Cosine Similarity Score: 0.596		
openssl exportpem to pfx without private key	11:14:36	0
openssl export password	11:22:11	0
extract private key form pfx	11:28:43	0
windows 2016 server request new certificate	11:43:20	0
windows 2016 server request new certificate 256 bit	11:45:04	0

*: The example sessions were fabricated to protect users' identity.
†: search queries that result in more ads in search result pages.

Our analysis of fraudulent traffic sessions revealed numerous sessions populated with unrelated queries, suggesting that fraudsters may use randomized keywords to diversify queries and obscure high-value search terms (i.e., queries that yield more ads for click manipulation), as shown by the highlighted rows in Table 1. In contrast, the organic sessions in Table 1 generally focus on specific topics, with queries directed toward relevant results. This observation led to the critical insight driving our research: from a search engine's perspective, *click fraud can be identified through features derived from consecutive search queries within each session*. We further define the difference in the query coherence across fraudulent and organic sessions as *session (in)coherence*.

2.4. Detection with Cosine Similarity vs. CoSeC

While an experienced fraud investigator may quickly identify fraudulent search queries, scaling this approach to flag fraudulent traffic automatically remains challenging. Search queries are in natural language, yet most detection algorithms perform best with numerical inputs, requiring

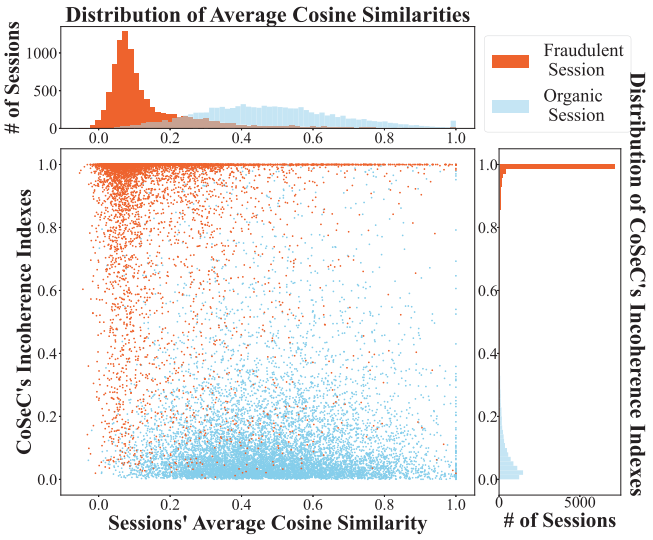

Figure 3: The Scatter and Histogram Distribution of Average Cosine Similarity (X-Axis) and CoSeC's Incoherence Indexes (Y-Axis) from Both Fraudulent and Organic Sessions. There is significant overlapping between fraudulent and organic sessions' average cosine similarity distribution, while CoSeC's incoherence indexes are clearly separated.

the encoding of queries into numerical scales or vectors. Fortunately, recent advancements in word- and sentence-level embeddings [23], [24] allow us to encode each query into a numerical vector. We then calculate the *cosine similarity* (CS) [25] between pairs of query vectors to quantify the similarity between two queries. For each session, we compute the average CS by averaging the similarities between consecutive queries.

The top histogram in Figure 3 shows the average CS of our preliminary dataset. A notable gap appears between the average CS values of organic and fraudulent sessions, with fraudulent sessions showing a significantly lower average CS. This aligns with our observation that fraudsters use randomized search queries to obscure high-value search terms. However, a significant overlap remains between organic and fraudulent sessions. In examining overlapping cases of fraudulent traffic, we found patterns similar to the fraudulent session 2 in Table 1. Fraudsters employed both similar (Rows 1 and 2) and repeated (Rows 3-5) search queries within a session, artificially inflating the average CS. This suggests that if search engines rely solely on sessions' average CS as a detection feature, fraudsters could evade detection by randomly repeating queries within a session.

The key insight is that even with a high average CS, queries in fraudulent sessions often lack logical coherence—a discrepancy that experienced investigators can identify. Human investigators rely on the fine-grained semantic (in)coherence among queries rather than the similarity metrics to detect fraudulent traffic. Such nuanced context is challenging for statistical algorithms and existing machine learning models to capture effectively. This challenge motivated the design of CoSeC, which integrates

TABLE 2: SYMBOLS & DEFINITIONS IN §3.

Symbol	Description	Type
Search Request Page (P)		
uid	User ID	ID
q	Search query string	string
t	Timestamp of the search request	timestamp
$tint$	Time Interval with previous request in second	int
$anum$	Number of Ads returned on the page	int
$cnum$	Count of ad clicks on the page	int
r	Revenue generated by ad clicks in this page	float
pn	Page number of the search result	int
ua	User-Agent	string
ip	IP Address	string
$\{AC\}$	Set of ad click requests on this search page	Set
Session (S)		
uid	User ID	ID
$\{P_i \mid i \in [1, l]\}$	Set of consecutive search pages	Set

literal semantics, temporal patterns, and ad-click behavioral information from search sessions to detect fraudulent traffic. CoSeC addresses the challenges of encoding these multidimensional features by sequentially interpreting each search request in a session. This enables CoSeC to capture both session-level context and individual request-level information, deriving the sessions' incoherence index, which reflects the session's overall coherence.

As a proof of concept, we applied CoSeC to the preliminary dataset to compute the incoherence index for each session. Combining this distribution with the average CS distribution, we plotted the scatter distribution in Figure 3, where each point represents a session's average CS (X-axis) and incoherence index (Y-axis). The histogram on the right illustrates the distribution of CoSeC's incoherence index, revealing a more distinct separation between organic and fraudulent sessions. This clear distinction allows search engines to set a decision boundary that flags fraudulent traffic with minimal false positives and false negatives. Next, we detail CoSeC's methodology in §3, CoSeC's automated dataset collection procedure in §4, and we evaluate CoSeC's performance in §5. We specifically compare the performance of CoSeC with machine learning models that use statistical features in §5.4.

3. Methodology

Overview. Motivated by the ultimate goal of detecting fraudulent activities through the incoherence of search sessions, Figure 4 presents the end-to-end pipeline of CoSeC in three distinct steps. First, CoSeC organizes and segments the search sessions from individual search requests (§3.1). Next, CoSeC extracts and encodes the multidimensional features using the *Multidimensional Semantic Feature Encoder* (§3.2). Finally, CoSeC feeds the encoded sequential feature vectors into the *Incoherence Index Evaluator* (detailed in §3.3) to predict the *incoherence index*. Symbols used throughout this section are summarized in Table 2 for reference.

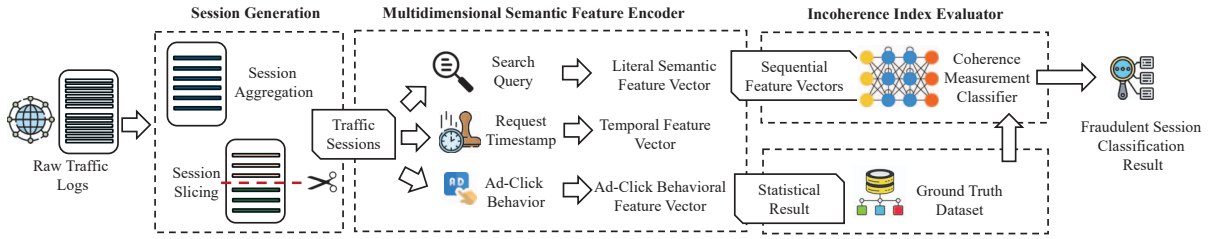

Figure 4: CoSEC's Pipeline Overview.

3.1. Session Generation

By taking the raw search and ad-click request logs as input, the initial step of CoSEC is to generate search sessions from these raw traffic logs. CoSEC generates sessions by tracking traffic with cookies enabled by the publisher's website. For session management, non-persistent session cookies are generally classified as "strictly necessary first-party cookies" and do not require user consent under GDPR [26]. As a result, organic users are generally advised to enable these first-party cookies on search websites. From a fraudster's perspective, traffic without valid cookie information is more likely to be flagged as abnormal or bot traffic. Consequently, fraudsters must mimic benign user behavior by setting valid cookies. We further discuss countermeasures against fraudsters who attempt to manipulate or invalidate cookies to disrupt CoSEC's session-generation methodology in §7.2.

Algorithm 1 summarizes CoSEC's session-generation procedures, comprising two steps: *session aggregation* and *session slicing*, as shown in Figure 4. CoSEC generates sessions from raw traffic logs, where the input is a list of search request pages, denoted by $Pages$. Additionally, CoSEC utilizes a predefined threshold T for session slicing. First, CoSEC sorts and groups the pages by the users' identifier (uid in Line 3 - Line 4). Next, CoSEC enumerates each grouped page list (in Line 5) and sorts the pages by the request's timestamp t (Line 6). At this point, CoSEC completes the *session aggregation* step by organizing and ordering pages for each user. For *session slicing*, CoSEC processes each search request page P in the list, obtaining ad-click behavioral information (Line 10) and calculating the time interval relative to the previous request (if one exists) in Line 14. If the time interval surpasses the threshold T, CoSEC divides the sequence of requests into two sessions (Line 11). Finally, CoSEC outputs all the sliced sessions Line 20.

Another challenge lies in determining the optimal timeout threshold for session slicing. CoSEC addresses this by conducting our additional preliminary measurement, as detailed in §A. As a result, CoSEC utilizes a 60-minute threshold, providing the best capability to measure the incoherence index for both fraudulent and organic sessions. Additionally, even when a session is sliced based on the time interval between two requests, search engines may still encounter cases where users, or more commonly bots, continue generating search requests, resulting in unusually long sessions. To manage such cases, CoSEC caps the session length at a maximum of 50 search requests. This cap is sufficient for CoSEC to accurately measure the incoherence index, as illustrated in §5.3.

Algorithm 1: Session Generation Algorithm

Input: $Pages$: List of search request pages;
Input: T: Threshold to slice page sequence into different sessions;
Output: $Sessions$: Sessions Generated From Raw Traffic Logs;

1 **Function** *GetSessionsFromPages*($Pages, T$)**:**
2 $Sessions = \phi$;
// Sort and group pages by uid
3 $Pages = Pages.sort(P \rightarrow P.uid)$;
4 $PagesByUID = Pages.groupBy(P \rightarrow P.uid)$;
5 **for** $Pages_{uid} \in PagesByUID$ **do**
// Sort user's pages by timestamp
6 $Pages_{uid} = Pages_{uid}.sort(P \rightarrow P.t)$;
// Process each user's page sequence
7 $S = \phi$;
8 $P_{tmp} = Null$;
9 **for** $P \in Pages_{uid}$ **do**
// Get ad-click features.
10 $AggregateAdClickFeatures(P)$;
// Slice the session when the time interval is larger than T
11 **if** $P_{tmp} \neq Null$ && $P.t - P_{tmp}.t > T$ **then**
12 $Sessions.add(S)$;
13 $S.empty()$;
// Get time interval between queries for each P
14 $P.tint = P.t - P_{tmp}$;
15 **end**
16 $P_{tmp} = P$;
17 $S.add(P_{tmp})$;
18 **end**
19 **end**
// Return the generated sessions
20 **return** $Sessions$;
21 **end**
22 **Function** *AggregateAdClickFeatures*(P)**:**
// Process each click request in P
23 **for** $c \in P.\{AC\}$ **do**
// Sum up revenue and count clicks.
24 $P.r+ = c.r$;
25 $P.cnum+ = 1$;
26 **end**
27 **end**

3.2. Multidimensional Semantic Feature Encoder

With the generated search sessions, the next step for CoSEC is to extract and encode raw multidimensional features into numerical representations that can be interpreted by the deep model. To ensure the general

applicability of CoSeC, we focus only on features that are accessible to most search engines and less susceptible to manipulation by fraudsters. Specifically, as shown in Figure 4, CoSeC leverages three groups of features from search sessions: (1) literal semantic features extracted from consecutive search queries (q), (2) the temporal features extracted from the consecutive requests' timestamps (t), and (3) the ad-click behavioral features gathered from ad-click requests, partially generated with Algorithm 1. These features capture user or bot interactions with the search engine and ad clicks. Although fraudsters may manipulate keywords, request timings, and selective ad-click logic, the data observed by search engines still reflects distinctive bot behavior. In contrast, features such as the IP address or user agent are easily and heavily manipulated by fraudsters, causing a disconnect between the data received by search engines and the actual bot activity. Consequently, CoSeC does not consider these features.

A fundamental challenge for CoSeC is extracting meaningful insights from the limited features accessible to search engines. As noted in §2.2, session-aggregated features (e.g., session length, request frequency) are too course-grained, lacking the granularity needed for CoSeC to access session incoherence accurately. To address this challenge, CoSeC extracts three types of features from each individual request and concatenates these feature vectors into a sequence as input. This approach enables CoSeC to capture session-level information by internally aggregating the request-level feature values with the deep model while also leveraging insights from incoherent requests within the session to produce the final results.

3.2.1. Literal Semantic Feature Encoding. To convert the search keywords into numerical form and reduce dimensionality, CoSeC embeds each entire search query into a fixed-length numerical representation independent of the query's length. Building on the concept of Sentence-BERT [24], CoSeC employs a SOTA transformer-based algorithm for query encoding tasks. CoSeC segments the search query q into individual words w_i by splitting at spaces and generates a numerical token for each word using the *Literal Semantic Tokenizer* ($\sigma_{literal}$). Next, CoSeC applies average pooling to compute the final literal features $\vec{F^l}$ from a query with count of m words as:

$$\vec{F^l} = \frac{1}{m}\sum_{i=1}^{m}\sigma_{literal}(w_i)$$

Notably, this technique also enables our preliminary investigation of search queries in §2.2. For a given query pair, q_1 and q_2, the query similarity is calculated using the CS of the encoded feature vectors, represented as:

$$similarity = \frac{F^l_{q_1} \cdot F^l_{q_2}}{|F^l_{q_1}| \times |F^l_{q_2}|}$$

The session-level average query similarity is the average of each consecutive query pair, calculated as:

$$session\ query\ similarity = \frac{1}{n-1}\sum_{i=1}^{n-1}\frac{F^l_{q_i} \cdot F^l_{q_{i+1}}}{|F^l_{q_i}| \times |F^l_{q_{i+1}}|}$$

where q_i is the i^{th} search query from a session with length n. This session-level average CS score reflects the semantic similarity between consecutive queries. As shown in Table 1, the organic session examples have scores of 0.687 and 0.596, respectively. The incoherent session example 1, which uses randomized queries, has a score of 0.089.

3.2.2. Temporal Feature Encoding. CoSeC extracts numerical representations from the timestamps t of the search requests within a session. Initially, CoSeC captures direct temporal information from each timestamp through *ad-hoc temporal feature encoding*. Specifically, for each search request, CoSeC extracts the hour, minute, and second as initial features. Next, CoSeC applies both normalization and cyclical encoding to each feature. For example, the hour value h is normalized to the range $[0, 1]$ as $h/24$ and then encoded into a cyclical representation as $\cos(2\pi h/24),\ \sin(2\pi h/24)$. This cyclical encoding allows the model to capture continuity across natural day transitions, where the hour value shifts from 23 to 0.

We use $\vec{t}$ to represent the numerical vector from the concatenation of both direct and cyclical encoding values. In addition to the direct values extracted from the timestamp, CoSeC also measures the time interval, as detailed in §3.1. The time interval, denoted as Δt in Table 2, is calculated in seconds and normalized by dividing it by 3,600, which is the timeout threshold for session slicing. Finally, CoSeC concatenates all these numerical values and then applies a single-layer autoencoder to derive temporal features $\vec{F^t}$ as:

$$\vec{F^t} = \sigma_{temporal}(concat(\vec{t}, \Delta t))$$

3.2.3. Ad-click Behavioral Feature Encoding. In addition to the literal and temporal information provided by query keywords and timestamps, CoSeC also captures features that represent ad-click behaviors in both organic and fraudulent sessions. This process encodes knowledge of ad-click characteristics within their associated search requests. CoSeC gathers the number of ads displayed on the search result page (a), the count of ad clicks (c), the revenue derived from ad clicks (r), and the current page number of the search results (pn). Notably, the page number (pn) reflects user behavior patterns when navigating through multiple search result pages, indicating a more persistent search behavior when initial results do not meet their expectations. This feature helps CoSeC distinguish between typical browsing behavior and potential bot activity. We refer to this step as *ad-hoc ad-click behavioral feature encoding*. Next, CoSeC applies an autoencoder, similar to the one used for temporal features, to the concatenated

vector of ad-click features. The final ad-click behavioral feature, $\vec{F^b}$, is represented as:

$$\vec{F^b} = \sigma_{behavioral}(concat(a, c, r, pn))$$

3.3. Incoherence Index Evaluator

Next, as illustrated in Figure 4, CoSeC's Incoherence Index Evaluator assesses the coherence of feature vectors extracted from consecutive search requests within a session. This task has two technical challenges: (1) the model must accommodate sessions of varying lengths, and (2) the model should introduce minimal computational overhead while ensuring robustness in measuring the incoherence index.

CoSeC employs a Bi-LSTM [27] as the backbone sequential model to process the extracted features and compute the incoherence index. This model enables CoSeC to handle feature vector sequences of arbitrary length, deriving a bidirectional, session-level intermediate representation. Although transformer-based models can process sequential data, we opt for a Bi-LSTM model as a proof of concept due to the relatively short sequence lengths in search sessions (i.e., mostly under 50, as shown in Figure 2), the limited size of the training dataset, and computational constraints. Specifically, for a given search session S with n search requests, CoSeC uses semantic feature extraction techniques to generate an initial input feature sequence, represented as $\vec{V} = \{X_1, X_2, ..., X_n\}$, where each $X_i = concat(\vec{F_i^l}, \vec{F_i^t}, \vec{F_i^b})$ represents the concatenation of the multidimensional features of the i^{th} search request P_i in a search session S. Following the state-of-the-art precedent [23], we prepend and append start and end tokens to $\vec{V}$ as $X_{[STR]}$ and $X_{[END]}$. Both tokens are generated as constant, randomly initialized, normally distributed vectors. For simplicity, despite the internal complexity of Bi-LSTM models, we denote the Bi-LSTM processing function as σ_{lstm}.

During both training and prediction, the Bi-LSTM model processes sequences in the forward and backward directions. For a given sequence in the forward direction, the input to the Bi-LSTM model includes two vectors: (1) the feature vector from a search request (X_i) and (2) a hidden vector representing the accumulated state from previous inputs ($X^{f-lstm}i-1$), where $f-lstm$ denotes Bi-LSTM in the forward direction. For each input feature vector, the Bi-LSTM model predicts the updated hidden vector ($X^{f-lstm}i$) as:

$$X_i^{f-lstm} = \sigma_{lstm}(X_i, X_{i-1}^{f-lstm})$$

In the backward prediction, the Bi-LSTM starts from the end of the sequence, and the hidden vector at each node's output X_i^{b-lstm} is derived as:

$$X_i^{b-lstm} = \sigma_{lstm}(X_i, X_{i+1}^{b-lstm})$$

In both directions, CoSeC initializes the hidden vector as a zero vector. To obtain the final output, CoSeC concatenates the backward hidden vector of the $[STR]$ token and the forward hidden vector of the $[END]$ token. A multilayer perceptron (MLP) then processes this concatenated output to produce the final prediction, $pred$, represented as:

$$pred = \sigma_{mlp}(concat(X_{[STR]}^{b-lstm}, X_{[END]}^{f-lstm}))$$

where σ_{mlp} represents the MLP, converting the bidirectional LSTM model's output vector into a scalar value ranging from 0 to 1. We refer to this output as the session's *incoherence index*.

Training Objective Task. Since no existing dataset provides ground-truth labels for incoherence measurement, we derive incoherence labels based on the inherent differences between organic and fraudulent sessions. To achieve this, we collect datasets containing both fraudulent and organic sessions, with our label derivation rules detailed in §4.1 Given that most fraudulent sessions are likely to be incoherent while organic sessions tend to be coherent, we simplify this task into a binary classification model trained to distinguish between fraudulent sessions (labeled as 1) and organic sessions (labeled as 0). CoSeC employs binary cross-entropy as the loss function $\mathcal{L}$. For N sessions in a training batch, the loss is represented as:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}(y_i \times log(pred_i) + (1-y_i) \times log(1-pred_i))$$

4. Ground-Truth Dataset Collection

We built our training dataset and evaluation benchmark from real-world search traffic. Dataset collection presents significant challenges in this research area, as no approach can definitively confirm whether a request originates from click fraud without input from the fraudsters. Existing studies [9], [12] rely on either collaborator-provided labels or labeling algorithms, making the dataset quality dependent on the existing fraud detector's accuracy. To minimize such bias, we develop three solid rules, as introduced in §4.1 to sample organic and fraudulent traffic from real-world data and to evaluate CoSeC's performance. Our evaluation starts with unlabeled traffic provided by our collaborator. To optimize our understanding and validation of CoSeC's performance, we focus on English-language sessions as a proof of concept, using [28] for language selection. The extension of CoSeC to sessions in other languages is discussed in §7.3.

Our datasets were created by filtering sessions from traffic in February and March and applying the three rules, resulting in a separate dataset for each rule (i.e., Rules R1-R3 in §4.1). We further validated the quality of these datasets in §4.2. The datasets derived from February and March are used for performance and drift evaluations, respectively, in §5. Notably, we ensure that the information used in dataset collection rules, such as user agent or publisher information, remains orthogonal to the features leveraged by CoSeC. This prevents sampling bias or label leakage into CoSeC. We further discuss potential limitations of our dataset collection procedures in §7.

TABLE 3: DATASET LABELING VALIDATION RESULTS.

Rule	Label	Dataset Size	# Sampled Sessions	#TP	#FP	Noise Ratio
R1-Dataset	0	200,000	500	484	16	3.20%
R2-Dataset	1	100,000	500	500	0	0.00%
R3-Dataset	1	100,000	500	486	14	2.60%
Total	-	400,000	1,500	1,470	30	2.00%

Additionally, we collected unlabeled traffic since April 2024 to evaluate CoSeC under real-world conditions, as demonstrated in the case studies presented in §6.

4.1. Ground-Truth Label Collection Rules

Organic Traffic Collection (R1). We collect our organic traffic dataset from search and ad-click requests on search engine-owned websites that fraudsters cannot profit from, making fraudulent activity unlikely. A recent study [29] also suggests that traffic from outdated browser versions is typically bot-generated, while organic users generally use auto-updated, latest versions. As a result, our collection process is as follows: (1) we gather traffic from search engine websites; (2) we generate sessions, filtering out those with inconsistent user agents or outdated browsers; and (3) we exclude sessions linked to known fraud campaigns based on collaborator-provided deny lists. This approach yields a representative set of organic traffic requests, capturing diverse user sessions across different countries, cultures, and demographics.

Sessions With Inconsistency Behaviors (R2). Search engines can identify fraudulent traffic by leveraging session-level inconsistencies as ground truth. In organic sessions, details like OS type, browser version, and browser name remain consistent throughout. In contrast, sessions with inconsistencies in these fields are likely due to request header manipulation or automated scripting [12]. We collect sessions with inconsistent user agents as indicators of fraudulent traffic, allowing search engines to detect suspicious activity that attempts to evade detection through request manipulation. This rule is designed to capture a high-precision fraudulent dataset, encompassing general fraudulent traffic from small to medium-scale fraudsters.

Sessions From Suspicious Channels (R3). Drawing on insights from [5], we identify fraudulent traffic from certain publishers by grouping search sessions by publisher ID and calculating the six statistical metrics outlined in §2.2. We compare these metrics with organic traffic (*R1*) and flag publishers whose metrics deviate most from the norm on a daily basis. Flagged publishers are sorted by traffic volume, and sessions from the top five are included in the fraudulent traffic dataset. Notably, such fraudsters are equipped with resources to actively adopt evasion techniques, allowing their traffic to bypass the dataset collection approach presented in R2. This approach effectively collects fraudulent traffic from sophisticated fraudsters while minimizing false positives, as validated in §4.2.

4.2. Label Correctness Validation

We prepared the datasets for training and performance evaluation by applying Rules R1–R3 to raw search traffic. From each dataset, we randomly sampled 500 sessions and manually verified their accuracy. Ground-truth labels were provided by experienced fraud investigators, authorized by our collaborator, who reviewed the samples without knowledge of the specific rule applied to each session. In the absence of direct data from fraudsters, this labeling method offers the most reliable ground truth and meets both academic and industry standards.

Table 3 presents the validation results: Columns 1 and 2 list the dataset collection rules and assigned labels. Column 3 shows dataset sizes, Column 4 indicates the number of sessions sampled for validation. Columns 5 and 6 display true positives (TP) and false positives (FP), and Column 7 shows the datasets' noise ratio.

As shown in Row 1 in Table 3, Rule R1 collected 200,000 sessions from organic users. Among 500 sampled sessions for manual validation, we identified 16 (3.2%) noisy sessions. These false positives, flagged as fraudulent by investigators, exhibited patterns similar to known fraudulent campaigns, characterized by nonsensical and repetitive queries that generated high ad-click volumes. In Row 2 in Table 3, none of the 500 sampled sessions filtered by R2 were classified as organic. These sessions demonstrated varied fraudulent behaviors. For example, in addition to the patterns shown in Table 1, fraudsters generated traffic with repeated keywords with random ad clicks. In other cases, fraudsters used the "site:" operator to retrieve results from a specific domain, appending random movie titles to each query. We also observed cases where fraudsters segmented articles into sentences for sequential searches, creating sessions with high average CS. Row 3 reports 14 noisy sessions (2.6% noise ratio), likely originating from organic users on the flagged publisher's website. Despite this noise, most sessions collected under R2 and R3 remain incoherent, making them well-suited for training and testing. In total, we manually verified 1,500 sampled sessions, identifying 30 noisy sessions. This results in an overall noise ratio of 2.0%. The validation demonstrates the robustness of our rules in capturing representative sessions for training and evaluating CoSeC.

5. Evaluation

5.1. Implementation & Evaluation Dataset Setup

We implemented CoSeC's prototypes in Python using PyTorch [30]. For the literal semantic feature encoder, we applied the method proposed in [24] with the *MPNet* implementation [31] and pre-trained weights from [32]. We used the AdamW optimizer [33] with a learning rate of 0.0001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 0.01. A step learning rate optimizer was employed with a step size of 2 and a gamma of 0.8. Training was conducted on a bare-metal machine with an 8-core CPU, 128GB of

memory, and a GPU with 16GB of memory. CoSEC was trained for 20 to 30 epochs, with early stopping to prevent overfitting. §C details CoSEC's overhead measurement.

Evaluation Benchmark. We used the datasets filtered by Rules R1-R3 from Feb. 1 to Feb. 14 for model training and performance evaluation. As shown in §4.2, we collected 200,000 sessions from R1-filtered traffic. We split them into a training dataset (140,000 sessions) and a testing dataset (60,000 sessions) using a 7:3 train-test split ratio. Similarly, we randomly sampled 100,000 sessions each from traffic filtered by R2 and R3, dividing them into training and testing sets at the same 7:3 ratio. In particular, the randomly sampled datasets from rules R2 and R3 share less than 1% overlapped sessions, and we remove them from the dataset sampled with R3 to avoid repeatedly using these sessions. For concept drift measurement in §5.5, we sampled traffic filtered by R1–R3 over four weeks in March, randomly selecting 40,000 sessions each week—20,000 from R1, 10,000 from R2, and 10,000 from R3. This evaluation dataset maintains the same distribution ratios regardless of the size used in §5.

5.2. CoSEC Performance and Noise Validation

We trained CoSEC's incoherence index evaluator with the training datasets and evaluated it with the testing datasets. As shown in Table 4, CoSEC achieves an overall accuracy of 94.17%, with a precision of 95.79%, recall of 92.40%, and F1-score of 0.9407. Specifically, CoSEC flags 28,047 fraudulent sessions out of 30,000 sessions filtered by R2, yielding an accuracy of 93.49%, and successfully identifies 91.31% fraudulent sessions labeled by R3. Among 60,000 organic sessions, CoSEC produces only 2,684 false positives, resulting in an accuracy of 95.53%. CoSEC achieves an AUC of 0.9829 for the Receiver Operating Characteristic (ROC) curve and an AUC of 0.9853 for the Precision-Recall curve, highlighting its robustness in identifying the majority of fraudulent traffic with minimal false positives.

Given the challenge of manually labeling the whole test dataset, we look into false-positive (FP) and false-negative (FN) sessions reported by CoSEC for each dataset to evaluate its performance on noisy data. For manual validation, we sampled 500 FP sessions from the R1 dataset and 500 FN sessions each from the R2 and R3 datasets. We conducted the same manual validation procedures detailed in §4.2. The validation results are presented in Table 5 to conclude CoSEC's performance on the noisy data in each dataset. Columns 1 and 2 list the datasets' names and sizes. Column 3 shows the noise ratio derived in previous labeling validation in §4.2. Column 4 presents the count of FP or FN sessions reported by CoSEC on the testing dataset, and Column 5 shows the CoSEC's accuracy against each dataset. Columns 6-8 represent the manual validation results on 500 sampled sessions. Column 6 shows the number of sessions considered noisy and on which sessions CoSEC performs correctly. Column 7 shows the number of sessions being correctly labeled and CoSEC-made mistakes. Column 8 is the ratio of noisy sessions in the sampled 500 sessions. We further present the estimated CoSEC performance on all noisy sessions in each dataset in Columns 9-12. We estimate the total noisy sessions in Column 9 by multiplying the Dataset Noise Ratio (Column 2) with the size (Column 3). Column 10 estimates the total noisy sessions flagged by CoSEC in each test dataset by multiplying the noise ratio in FP/FN sessions (Column 8) by the count in Column 4. Column 11 estimates the missed noisy sessions, and Column 12 reports the percentage of noisy sessions being identified by CoSEC over total estimated noisy session counts.

TABLE 4: CoSEC's PERFORMANCE METRICS.

Rules*	# Session	#TP†	#TN†	#FP†	#FN†	Accuracy
R1-Dataset	60,000	0	57,316	2,684	0	95.53%
R2-Dataset	30,000	28,047	0	0	1,953	93.49%
R3-Dataset	30,000	27,393	0	0	2,607	91.31%
Total	120,000	55,440	57,316	2,684	3,560	94.17%

Precision: 95.79% **Recall**: 92.40% **F1-Score**: 0.9407
AOC$_{ROC}$: 0.9829 **AOC**$_{Precision-Recall}$: 0.9853

*: Rules represent the datasets being collected with.
TP: Fraudulent sessions correctly flagged by CoSEC.
TN: Organic sessions correctly not flagged by CoSEC.
FP: Organic sessions wrongly flagged by CoSEC.
TP: Fraudulent sessions wrongly not flagged by CoSEC.

With a noise ratio of 3.2% in the R1 Dataset, we estimate there are approximately 1,920 noisy sessions. Additionally, as shown in Table 5, 318 out of 500 (63.6%) reported false-positive (FP) sessions were identified as noise, suggesting that an estimated 1,707 noisy sessions out of 2,684 FPs are actually fraudulent but pass the filtering rule R1. Despite potential deviations in these estimates, they underscore CoSEC's ability to flag around 88.91% of fraudulent sessions in the R1 Dataset, capturing fraudulent sessions missed by the rule-based approach. This result overall remains consistent with CoSEC's performance on fraudulent traffic flagged by Rules R2 and R3. Notably, as elaborated in §4.1, these noisy sessions originate directly from client devices to search engine-owned publisher sites, generating no direct profit for fraudsters through an untrusted search interface (as depicted in Figure 1). Techniques that rely on flagging unethical publishers would typically overlook such sessions. However, these sessions still benefit fraudsters, as they may aim to "poison" organic traffic from trusted publishers, reducing the statistical feature gap between organic and fraudulent traffic. Fortunately, CoSEC can still flag 88.91% of these sessions as fraudulent at a fine-grained level, demonstrating its ability to detect previously unknown fraudulent activity without training on campaign-specific traffic.

The R2 dataset contains no noise, as it only includes sessions with forged and inconsistent user agents. Applying a similar analysis to the R3 dataset, we manually identified 129 noisy sessions out of 500 sampled sessions, which were subsequently considered organic. This translates to an estimated 780 noisy sessions in the R3 dataset, with CoSEC successfully flagging approximately 673 of them, resulting in an 86.28% identification rate for noisy sessions. Although CoSEC's performance on noisy sessions is

TABLE 5: MANUAL VALIDATION ON COSEC'S PERFORMANCE AGAINST NOISE.

Dataset	Size	Dataset Noise	CoSeC Results		Manual Val. in Samplied FP/FN			Est. CoSeC's Performance on Noise			
		Ratio(%)[1]	#FP/FN[2]	Accu.	#Noise[3]	#Err[3]	%Noise[3]	#Total[1]	#Flg.[4]	#Mis.[4]	%Noise Flg.[5]
R1	60,000	3.20%	2,684	95.53%	318	182	63.6%	1,920	1,707	213	88.91%
R2	30,000	0.00%	1,953	93.49%	0	500	0.0%	0	0	0	N/A
R3	30,000	2.60%	2,607	91.31%	129	371	25.8%	780	673	107	86.28%
Total	120,000	2.25%	7,244	94.17%	447	1,053	29.80%	2,700	2,380	320	88.15%

1: Noise ratios are from labeling validation (§4.2). **#Total** estimates count noisy sessions by multiplying the noise ratio with dataset sizes.
2: Count of CoSeC-reported false-positive sessions in R1 dataset and false-negative sessions in R2 and R3 datasets regardless of noise.
3: The manual validation results from 500 sessions sampled from FP/FN results in each dataset. **#Noise** shows noisy sessions counts and **#Err** shows counts of sessions are actual wrongly labeled by CoSeC. **%Noise** shows the ratio of noisy sessions in sampled FP/FN sessions.
4: Estimated counting of noisy sessions' successfully flagged by CoSeC. **#Flg.** values are estimated by **#FP/FN of CoSeC Results** (Column 4) multiplies **%Noise** of FP/FN sessions (Column 8). Results are rounded to integers.
5: Percentage of estimated noisy sessions in each dataset being successfully flagged by CoSeC.

evaluated to the best of our knowledge, these results demonstrate CoSeC's robustness, with an overall success rate of 88.15% in correctly identifying noise.

5.3. Feature-Relevant Evaluation

We evaluated CoSeC's performance by selectively masking features during training and testing to assess the impact of different input features. We categorized the features into three groups, as defined in §3.2. Additionally, we assessed overall accuracy by training the prototype on datasets ranging from 4,000 to 400,000 sessions, covering 1% to 100% of the evaluation datasets. As in §5.2, the prototype was trained on 70% of each dataset and tested on the remaining 30% at each dataset scale. Figure 5 shows the accuracy results based on individual or combined feature groups, resulting in seven curves.

In general, CoSeC achieves optimal performance with 94.17% accuracy when all feature groups are utilized. As shown when the dataset has 400,000 sessions, CoSeC achieves 91.81% accuracy using only literal features, 92.96% when combining literal and temporal features, and 92.61% when combining literal and behavioral features. These results demonstrate the robustness of literal features in identifying fraudulent traffic. In contrast, CoSeC trained with temporal features achieves an accuracy of 86.72%, while using only ad-click behavioral features results in a maximum accuracy of 71.72%. Even combining both temporal and ad-click behavioral features, the accuracy can only achieve 88.11%, which is 6.06% lower than that including additional literal features, highlighting the limitations of these feature types for click fraud detection. However, as shown by the solid line curve in Figure 5, combining all three feature groups consistently delivers the best performance in all dataset sizes evaluated.

In terms of dataset sizes, the curves in Figure 5 show that CoSeC achieves better performance with larger training datasets. Additionally, the results reveal that temporal features capture a clearer distinction between organic and fraudulent sessions when trained on smaller datasets (i.e., total dataset size below 100,000 sessions). In contrast, CoSeC 's prototypes utilizing literal features require more data to detect this gap effectively. For

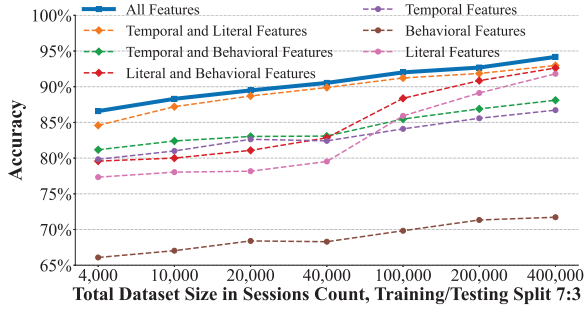

Figure 5: CoSeC's Accuracy by Different Features Selected and on Different Dataset Sizes. Sizes are the total dataset size, split into train/test sets at 7:3.

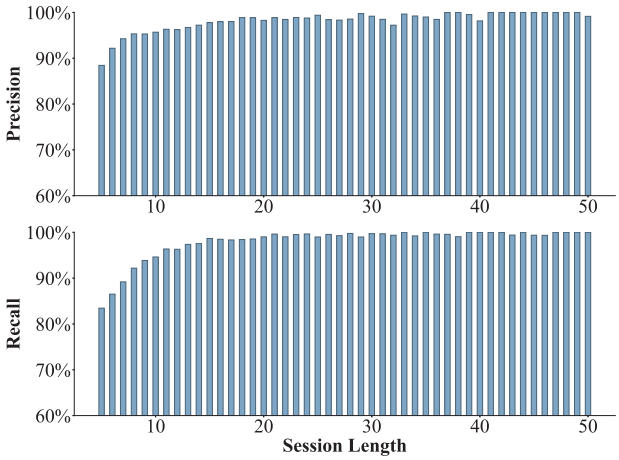

Figure 6: Precisions And Recalls by Session Lengths.

example, models trained with only literal features achieve accuracy of 85.91% with 100,000 sessions and 79.52% with 40,000 sessions, showing a gain of 6.39%. However, the prototype using only temporal features shows a smaller gain, improving from 82.41% to 84.09%, with an increase of just 1.68%. Given that search engines can access large-scale real-world data, we expect literal features to play an increasingly significant role in detecting click fraud in search ads.

Performance vs. Session Lengths. We further evaluated CoSeC's performance across varying session lengths.

TABLE 6: COMPARISON BETWEEN CoSeC AND STATISTICAL FEATURES-BASED MODELS.

Model	Accuracy	Precision	Recall	F1	AUC_{ROC}
SO-DNN	80.54%	88.46%	70.25%	0.7831	0.8745
SO-LR	71.42%	75.33%	63.71%	0.6903	0.7665
SO-RF	86.22%	90.95%	80.45%	0.8538	0.9247
S-DNN	80.61%	86.59%	72.44%	0.7888	0.8778
S-LR	77.40%	80.24%	72.71%	0.7629	0.8442
S-RF	90.35%	94.89%	85.29%	0.8983	0.9551
CoSeC	**94.17%**	**95.79%**	**92.40%**	**0.9406**	**0.9855**

SO-: models use all but no query similarity-related features.
S-: models use all statistical features.
LR: Logistic regression baseline model.
DNN: Deep neural network baseline model.
RF: Random forest baseline model.

Figure 6 presents CoSeC 's precision and recall breakdown by session length. Generally, both precision and recall improve with longer sessions, indicating CoSeC's enhanced capability to detect incoherent behaviors in extended sessions. Specifically, for sessions with 10 requests or fewer, the prototype achieves an average precision of 93.53% and recall of 89.87%. For sessions longer than 20 requests, CoSeC reaches an average precision of 99.63% and recall of 99.89%, reflecting improvements of 6.10% and 10.02%, respectively. This supports our initial assumption that session-level coherent context contributes to more accurate detection of fraudulent traffic.

5.4. CoSeC Versus Statistical Feature Models

Experiment Setup. To effectively evaluate the performance of CoSeC, we compared it with models leveraging two statistical feature setups. Guided by [4], we extracted 18 statistical features from the same raw session data used by CoSeC. The collection of these features is detailed in §B. Notably, while previous studies consider session-level information, none have utilized insights from query content or query similarities. We added nine additional features, shown in the last three rows of §B, resulting in a total of 27 features. We prepared two feature sets, as shown in Column 1 of Table 6: models with the *S-* prefix use all statistical features, while *SO-* models exclude nine query-based features, representing the standard statistical features available to search engines in studies summarized by [4]. We implemented deep neural networks (i.e., *DNN* models) with three fully connected layers, logistic regression (i.e., *LR* models), and Random Forest (i.e., *RF* models) as our baselines. All of these models produce an output representing an incoherence index in the range $[0, 1]$. We apply a threshold of 0.5, classifying sessions with predicted incoherence indexes above it as fraudulent.

Performance. Each row in Table 6 presents the accuracy, precision, recall, F1-score, and area-under-curve (AUC) values for Receiver Operating Characteristic (ROC) curves. Additionally, Figure 7 presents all models' ROC and

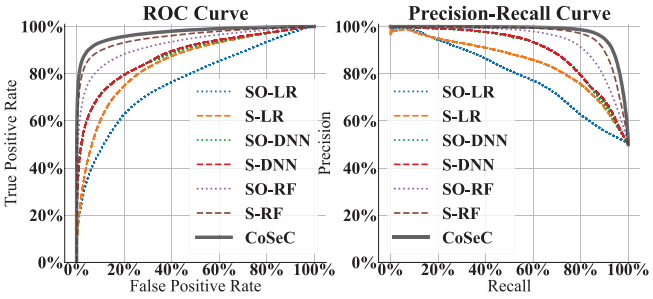

Figure 7: Receiver Operating Characteristic (ROC) Curve and Precision-Recall Curve for CoSeC and Models Using Statistical Features in Table 6.

TABLE 7: CONCEPT DRIFT MEASUREMENT RESULTS.

Dataset Period	Accuracy	Precision	Recall	F1-Score	AUC
Week 1	90.48%	94.37%	87.11%	0.9060	0.9685
Week 2	93.45%	93.08%	94.00%	0.9354	0.9789
Week 3	92.57%	94.23%	90.84%	0.9250	0.9786
Week 4	92.95%	94.08%	91.68%	0.9286	0.9740
Total	92.34%	93.94%	90.82%	0.9235	0.9750

Precision-Recall curves. Overall, CoSeC achieves the best performance, with a 94.17% accuracy, 95.79% precision, and 92.40% recall. This also leads to the best ROC and Precision-Recall curves with the maximum AUC_{ROC} of 0.9829. In terms of models running a similar computing overhead using all statistical features (i.e., S-DNN, S-LR, and S-RF), Random Forest achieves the best follow-up performance with 90.35% accuracy. This aligns with a previous study [34] highlighting that tree-based models are still superior in tabular data classification tasks. However, since CoSeC can more effectively leverage knowledge from the search requests, it still outperforms the *S-RF* model with gains of 3.82% in accuracy and 0.0304 in AUC_{ROC}. Additionally, by comparing models with and without query-similarity features, S-RF, S-DNN, and S-LR outperform SO-RF, SO-DNN, and SO-LR models with 4.32%, 5.98%, and 0.07% in accuracy, respectively. This finding sheds light on using query literal-semantic features to improve existing click fraud detection algorithms.

5.5. Concept Drift Measurement

To evaluate the reliability and robustness of CoSeC, we tested the prototype on datasets collected over four weeks, from March 1 to March 28. Since CoSeC was trained solely on February data, it had no prior knowledge of the March sessions. In Table 7, we present both session-level and individual search request-level evaluations for datasets collected each week in March. For each sub-dataset, we randomly sampled 40,000 sessions, comprising 20,000 organic and 20,000 fraudulent sessions, following the test dataset approach detailed in §5.1. Each row in Table 7 represents test results for a sub-dataset, with Column 1 indicating the dataset collection period. Columns 2–5 show accuracy, precision, recall, and F1-score,

respectively, while Column 10 shows the AUC value under the ROC curve for each dataset.

Comparing session-level evaluation results with those from the February dataset, the accuracy remains stable, averaging 92.34%. Precision, which measures the proportion of true positives among all positive detections, averages 93.94% $\pm$ 0.86%. The result remains close to the precision achieved in the initial performance evaluation. However, recall, tested across different datasets, shows more fluctuation, with an average of 90.82% $\pm$ 3.71%. Upon investigation, we attribute the stable precision to the relatively consistent search habits of organic users, which results in a low false-positive rate when CoSEC infers this traffic. In contrast, recall variability is likely caused by fraudsters' evolving evasion strategies. As search engines continually deploy new detection algorithms, fraudsters adapt their traffic generation tactics to bypass existing systems, leading to changes in fraudulent traffic patterns over time, which can impact the performance of CoSEC.

Nevertheless, even as fraudsters adopt new evasion approaches, CoSEC is able to flag 90.82% of fraudulent traffic with high precision (93.94%). Given CoSEC's reasonable overhead (as shown in §C), search engines can deploy CoSEC's methodology and dataset collection procedures to fine-tune the model continuously with newly identified fraudulent sessions.

6. Real-World Detection Case Studies

6.1. Case Study 1: Evolving Evasion Techniques

During our deploying CoSEC to real-world unlabeled traffic, we observed that a large group of identical search sessions flagged by CoSEC likely originated from the same campaign, which evolved its evasion strategies over time. Specifically, we noted that the campaign attempted to mimic mobile device traffic by manipulating user agents. Tracing back to February, these sessions fell within our detection scope because they featured multiple different mobile user agents within a single session. However, once flagged and invalidated, the attackers adjusted their approach by maintaining a consistent mobile user agent throughout each session, effectively circumventing the previous detection rule. During the seven days from April 1 to April 7, CoSEC identified 299,727 unique user agents within these sessions, reinforcing our initial assumption that fraudsters continually adapt their strategies to evade detection.

In terms of monetization and evasion, we observed that the campaign used 18 different publisher IDs, with each session containing requests from an average of 4.62 unique publisher IDs. These sessions were primarily redirected from nine different referrers, with each session originating from an average of 3.01 unique referrers. This strongly suggests that these sessions are manipulated by scripts, indicating a highly suspicious monetization channel through certain publishers involved in this fraud campaign. Regarding search queries, the fraud campaign employed a randomized keyword selection strategy, resulting in sessions similar to incoherent session example 1 shown in our preliminary study in Table 1. This pattern allows CoSEC to easily flag these sessions with a high incoherence index.

TABLE 8: SESSIONS WITH "SIMILAR" QUERIES.

Queries	Interval[1]	#Ads[2]	#Clicks[2]	Cos. Sim.[3]
downhole logging tools	-	0	0	-
downhole tool companies	123	0	0	0.644
downhole tools pdf	147	4	1	0.605
downhole tools manufacturers	43	0	0	0.740
downhole drilling tools	60	6	0	0.622
downhole tools	81	3	0	0.848
downhole completion tools	58	3	1	0.829

The example sessions were fabricated to protect users' identity.
1: The time interval between this request to the previous one in seconds.
2: Count of ads returned in search results and count of ads clicks.
3: Cosine similarity between this query to query of the previous request.

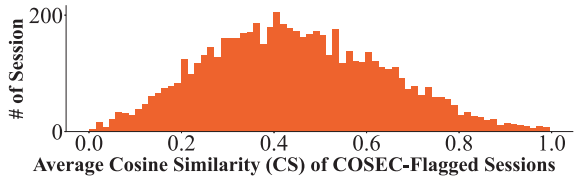

Figure 8: Cosine Similarity Distribution From CoSEC Flagged Sessions to Single Website in §6.2. As the fraudsters manipulate the queries, the query cosine similarities are similar to those from organic users' search sessions.

In a post-analysis, we collected all sessions generated by the campaign over a seven-day period. The campaign produced approximately 111,000 fraudulent sessions with clicks each day. Comparing these post-analysis findings with CoSEC's flagging results, we found that CoSEC successfully identified 87.65% of the search sessions within the collected traffic, underscoring its capability to detect fraudulent sessions even as fraudsters adapt their strategies. For flagged sessions, search engines can mitigate such attacks by canceling payments to the identified fraudulent traffic and discontinuing ad displays to these sessions.

6.2. Case Study 2: Search Query Manipulation

Not only are we investigating the manipulation of search query keywords to mimic organic user behavior, but fraudsters are also actively exploring this tactic. Between Nov. 1 and 10, CoSEC flagged 6,348 sessions generated on a third-party publisher's website, accounting for 8.89% of all search traffic to the site. After further investigation, our collaborator confirmed that these sessions were indeed fraudulent. Interestingly, we observed a clear trend in these flagged sessions toward using queries that mimic human thought processes. For example, in a fabricated search session shown in Table 8, besides the repeated keyword "downhole," other words are used to modify the query. These variations not only yield different search results but also create high similarity among queries, resulting in an average CS of 0.715. Additionally,Figure 8 shows the CS distribution for queries in these flagged sessions, closely

resembling that of organic search sessions in the top histogram of Figure 3.

Our investigation found that fraudsters may be using *suggested search queries* to build their lists. Specifically, as shown in Figure 9, modern search engines prompt users with potential queries as they type, a feature intended to aid organic users. Fraudsters appear to leverage this feature to create query lists that better mimic organic behavior. Fortunately, even as fraudsters adopt these new evasion techniques, CoSeC can still effectively flag these fraudulent sessions at scale.

7. Discussion

CoSeC's core objective is to mitigate click fraud in search engines. As outlined in §3, CoSeC uses general features like request timestamps, search keywords, and ad-click behaviors, which are commonly available across search engines. Search engines can also easily group requests by unique users, enabling efficient session construction from traffic. CoSeC is designed to flag fraudulent traffic retrospectively, enabling search engines to refund advertisers after identifying fraudulent traffic. Additionally, search engines may enhance fraud detection by combining CoSeC's output with other features.

7.1. Adversarial Attackers

Ad-click fraudsters continuously adapt their strategies to evade detection. However, since CoSeC measures session incoherence based on literal, temporal, and ad-click behavioral features, it remains resilient to common traffic manipulation techniques. Even if advanced fraudsters attempt to create more coherent sessions with high query similarity, as shown in §6.2, CoSeC still establishes a clear decision boundary, filtering out most fraudulent traffic. While adversaries might train a similar model for evasion, generating realistic search sessions that maximize illegal profit remains significantly more challenging. Since CoSeC leverages multidimensional patterns that remain consistent across sessions, attackers must carefully manipulate semantics, timing, and ad-click behavior rather than merely crafting keyword-based queries (as demonstrated in §6.2). Besides, slowing the rate of fraudulent traffic to evade CoSeC would significantly reduce fraudsters' profits, rendering click fraud economically unsustainable due to the costs of hiring botnets or click farms.

Although it is still possible, this task becomes more complicated when fraudsters target high-value search queries for higher profit. However, adaptive adversaries, who may abuse state-of-the-art black-box or white-box access to CoSeC's model to reverse-engineer CoSeC's detection criteria, will be an increasing challenge. Besides, the fast-paced evolution of LLM-based generators may enable fraudsters to develop a more comprehensive system to simulate organic users' personas and generate search and ad-click traffic. Despite the increasing operational cost to incorporate LLM-based generators and the technical challenge of overcoming existing misuse detections [35], we consider this a future research direction to improve CoSeC to defend such attacks.

7.2. Cookie Availability & Integrity

Fraudsters may attempt to manipulate cookies to disrupt CoSeC's session generation. However, ensuring cookie integrity is an orthogonal challenge managed by web service providers. Techniques such as signing cookies with a server-side key (e.g., DBSC [36]) make forging valid cookie values technically difficult. Besides, while disabling cookies could interfere with CoSeC's session generation, such traffic is likely flagged by search engines and has minimal impact on organic traffic revenue. Notably, privacy-conscious users who disable cookies may inadvertently blend their traffic with trivial fraudsters' traffic. Nonetheless, search engines should flag such ad-click traffic to prevent advertisers from being overcharged, even at the cost of losing some revenue from legitimate users. Additionally, industry-standard cookieless tracking methods [37] offer viable alternatives for grouping individual requests into session-level traffic, aligning with our research.

7.3. Limitation

Dateset Coverage. We acknowledge that our dataset does not encompass all search traffic and may introduce bias due to the absence of an automated method for deriving ground truth from blended data. Although we apply combined rules to filter fraudulent traffic from various sources, these rules may not span the entire domain of search behavior. As a result, CoSeC's performance could be affected when detecting fraudulent traffic across blended sources. This challenge is not unique to our research but is a common limitation in building and evaluating comprehensive machine learning-based ad-click fraud detection systems. However, this limitation underscores the necessity of our research in advancing fraudulent traffic detection. Furthermore, based on our proof-of-concept evaluation, we believe CoSeC's accuracy could be further improved with a more comprehensive and precise data source.

Technical Limitations. As a proof of concept, CoSeC has certain technical limitations. First, due to resource constraints, we rely on pre-trained sentence embedding models to encode search keywords, meaning the quality of semantic features depends on these models' performance. Additionally, performance may vary across different languages. Fine-tuning these models with search engine-specific datasets is a potential future enhancement. Another limitation is that parts of CoSeC's evaluation rely on assessments from click fraud investigators, as neither we nor our collaborators can definitively verify a request's legitimacy with the sender (i.e., fraudsters). Although our results align with industry standards and best practices, a margin of error remains.

8. Ethical Consideration

We have carefully considered the ethical implications of this research and conducted it responsibly, ensuring no user data was disclosed without explicit permission from our collaborators. We recognize the potential risks associated with re-identification and unethical model inversion attacks. To mitigate these risks, we implemented strict safeguards, including conducting all experiments on collaborator-owned devices that are not accessible from the public internet. Additionally, since CoSeC's output is used by our collaborator to either charge or not charge for an ad click, the only externally observable output is a binary decision for each ad click. This makes it highly unlikely to reconstruct the original user information from the output. These measures ensure that our research aligns with best privacy practices while safeguarding the integrity and confidentiality of user data.

In this research, we define sensitive information as any data that could be used to attribute behaviors to individuals, including search queries, IP addresses, device details such as user agent and OS type, and user IDs. Given the dataset's direct connection to search traffic between the collaborator's search engine and end users, all data access was classified as sensitive and restricted to company-owned and managed devices. Access to this data was limited to employees of our collaborator who are bound by Non-Disclosure Agreements and privacy-related employment regulations. Other researchers on this project did not have access to such sensitive data. To illustrate concepts without compromising privacy, only fabricated examples are used in Table 1 and Table 8.

9. Related Work

Server-side Click Fraud Detection. Several studies [7], [8], [22], [38] have successfully identified fraudulent traffic through machine learning algorithms based on traffic fingerprint features. EvilHunter [12] identifies and clusters fraudulent mobile devices, specifically click farms, based on ad bid behavior features. However, this approach is limited to traffic from manipulated bots, and attackers can evade detection by altering device information. ViceROI [5] and HK-Index [39] propose detecting fraudulent traffic through aggregated knowledge but face limitations when monetizable ad clicks are hidden within large volumes of organic traffic or when search engines require more granular fraud detection. Xu et al. [40] showed success in identifying fraud by monitoring behavior on advertisers' websites, though scalability remains challenging in practice. Our work builds on these ideas to create a ground-truth dataset for training and testing models. Unlike previous studies, CoSeC identifies fine-grained fraudulent traffic sessions based on semantic and temporal patterns, which are readily available to search engines.

Frontend Click Bot Remediation. In the realm of front-end click-bot remediation, studies have shown promising success in analyzing bot behavior and detecting fraudulent activity. Recent works [14], [15], [19], [41]–[43] can identify and remediate auto-click behaviors in mobile malware. For example, Kim et al. [41] distinguished ad clicks from user-generated and malicious sources by analyzing method call stack traces, while ClickScanner [16] utilized static features and Variational AutoEncoders to classify fraudulent mobile apps. Paul et al. [19] examined the ZeroAccess botnet's ad-click fraud mechanisms, and AdCube [44] highlighted risks and mitigations for ads placed outside users' view in VR devices. Additionally, research in mobile malware forensics and malware binary analysis—focused on evaluating behavioral patterns and identifying malicious traffic [45]–[50]—offers valuable insights into understanding and characterizing bot-generated traffic. Orthogonal to these studies, which focus on malicious bots, CoSeC addresses click fraud in search ads using generalizable features accessible to search engines, emphasizing the remediation of fraudulent traffic rather than bot behavior alone.

Anomaly Network Traffic Analysis. CoSeC is inspired by research addressing anomaly detection in network traffic. State-of-the-art techniques [51]–[56] have made significant progress in detecting fraud in banking and e-commerce transactions. For example, Wang et al. and Branco et al. [52], [53] leveraged consecutive user-behavior patterns to identify fraud. Li et al. [10] detected script-generated network sessions by fingerprinting HTTP headers. Unlike these studies, CoSeC tackles a fundamentally different task by using fine-grained semantic knowledge to identify fraudulent sessions We believe CoSeC's approach could extend to other types of anomaly analysis. Additionally, research by [57] highlights how understanding bot-generated query intent and behavior is essential, emphasizing the role of semantic knowledge from search sessions in detecting fraud.

10. Conclusion

In this study, we introduced CoSeC as the pioneering work in measuring coherence from search sessions to infer ad-click fraudulent activities. CoSeC relies exclusively on fundamental, invariant multidimensional features extracted from consecutive search requests to assess incoherence and profile user activity. As a proof-of-concept, we developed CoSeC's prototype and trained its classifier using raw real-world data, achieving 95.79% precision and 92.40% recall in identifying incoherent search sessions. Our results demonstrate CoSeC's promising performance in identifying incoherence within fraudulent sessions. Through evaluation of real-world datasets, we establish that this technique remains robust in countering the ever-evolving cheating strategies employed by fraud campaigns.

Acknowledgments

We thank the anonymous reviewers for their constructive comments and feedback. We also thank our

collaborators at Microsoft for their support, insights, and suggestions throughout this research. This material was supported in part by the Office of Naval Research (ONR) under grants N00014-19-1-2179 and N00014-23-1-2073; the National Science Foundation (NSF) under grant 2143689; and the Defense Advanced Research Projects Agency (DARPA) under contract N66001-21-C-4024. Any opinions, findings, and conclusions in this paper are those of the authors and do not necessarily reflect the views of our sponsors and collaborators.

References

[1] *Search advertising - united states*, https://www.statista.com/outlook/dmo/digital-advertising/search-advertising/united-states, [Accessed: 2024-04-11].

[2] *What is click fraud? | how click bots work*, https://www.cloudflare.com/learning/bots/what-is-click-fraud/, [Accessed: 2023-09-14].

[3] *The ultimate list of click fraud statistics 2023*, https://cheq.ai/blog/click-fraud-statistics-2023, [Accessed: 2023-09-14].

[4] R. A. Alzahrani and M. S. Aljabri, "AI-based techniques for ad click fraud detection and prevention: Review and research directions," *J. Sens. Actuator Networks*, vol. 12, no. 1, p. 4, Feb. 2023.

[5] V. Dave, S. Guha, and Y. Zhang, "ViceROI: Catching click-spam in search ad networks," in *Proceedings of the 20th ACM Conference on Computer and Communications Security (CCS)*, Berlin, Germany, Oct. 2013.

[6] V. Dave, S. Guha, and Y. Zhang, "Measuring and fingerprinting click-spam in ad networks," in *Proceedings of the 2012 ACM SIGCOMM*, Helsinki, Finland, Aug. 2012.

[7] N. Sahllal and E. M. Souidi, "Forecasting click fraud via machine learning algorithms," in *Proceedings of the 4th International Conference on Codes, Cryptology and Information Security*, Rabat, Morocco, May 2023.

[8] P. K. Keserwani, V. Jha, M. C. Govil, and E. S. Pilli, "Clickedroid: A methodology based on heuristic approach to detect mobile ad-click frauds," in *Proceedings of the International Conference on Paradigms of Computing, Communication and Data Sciences (PCCDS)*, Kurukshetra, India, May 2020.

[9] T. Graepel, J. Q. Candela, T. Borchert, and R. Herbrich, "Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine," in *Proceedings of the 27th International Conference on Machine Learning (ICML)*, Haifa, Israel, Jun. 2010.

[10] X. Li, B. A. Azad, A. Rahmati, and N. Nikiforakis, "Good bot, bad bot: Characterizing automated browsing activity," in *Proceedings of the 42nd IEEE Symposium on Security and Privacy (S&P)*, San Francisco, CA, May 2021.

[11] M. Kantardzic, C. Walgampaya, R. Yampolskiy, and R. J. Woo, "Click fraud prevention via multimodal evidence fusion by dempster-shafer theory," in *Proceedings of 2010 IEEE Conference on Multisensor Fusion and Integration (MFI)*, 2010.

[12] S. Sun, L. Yu, X. Zhang, M. Xue, R. Zhou, H. Zhu, S. Hao, and X. Lin, "Understanding and detecting mobile ad fraud through the lens of invalid traffic," in *Proceedings of the 28th ACM Conference on Computer and Communications Security (CCS)*, Seoul, South Korea, Nov. 2021.

[13] J. Szurdi, M. Luo, B. Kondracki, N. Nikiforakis, and N. Christin, "Where are you taking me?understanding abusive traffic distribution systems," in *Proceedings of the 30th International World Wide Web Conference (WWW)*, Virtual Conference, Apr. 2021.

[14] Z. Cai, Y. Nan, X. Wang, M. Long, Q. Ou, M. Yang, and Z. Zheng, "DARPA: combating asymmetric dark UI patterns on android with run-time view decorator," in *Proceedings of 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Network (DSN)*, Porto, Portugal, Jun. 2023.

[15] J. Crussell, R. Stevens, and H. Chen, "MAdFraud: Investigating ad fraud in android applications," in *Proceedings of the 12th ACM International Conference on Mobile Computing Systems (MobiSys)*, Bretton Woods, NH, Jun. 2014.

[16] T. Zhu, Y. Meng, H. Hu, X. Zhang, M. Xue, and H. Zhu, "Dissecting click fraud autonomy in the wild," in *Proceedings of the 28th ACM Conference on Computer and Communications Security (CCS)*, Seoul, South Korea, Nov. 2021.

[17] C. M. R. Haider, A. Iqbal, A. H. Rahman, and M. S. Rahman, "An ensemble learning based approach for impression fraud detection in mobile advertising," *J. Netw. Comput. Appl.*, vol. 112, pp. 126–141, Feb. 2018.

[18] *List of search engines*, https://en.wikipedia.org/wiki/List_of_search_engines, [Accessed: 2024-04-11].

[19] P. Pearce, V. Dave, C. Grier, K. Levchenko, S. Guha, D. McCoy, V. Paxson, S. Savage, and G. M. Voelker, "Characterizing large-scale click fraud in ZeroAccess," in *Proceedings of the 21st ACM Conference on Computer and Communications Security (CCS)*, Scottsdale, AZ, Nov. 2014.

[20] *It's not just websites, google search partner fraud has infiltrated the app ecosystem*, https://www.bandt.com.au/its-not-just-websites-google-search-partner-fraud-has-infiltrated-the-app-ecosystem/, [Accessed: 2024-04-11].

[21] S. Almahmoud, B. Hammo, and B. Al-Shboul, "Exploring non-human traffic in online digital advertisements: Analysis and prediction," in *Computational Collective Intelligence - 11th International Conference, (ICCCI)*, Hendaye, France, Sep. 2019.

[22] D. Yang, Z. Li, X. Wang, K. Salamatian, and G. Xie, "Exploiting the community structure of fraudulent keywords for fraud detection in web search," *Journal of Computer Science and Technology*, vol. 36, no. 5, pp. 1167–1183, Sep. 2021.

[23] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Minneapolis, MN, Jun. 2019.

[24] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing(EMNLP-IJCNLP)*, Hong Kong, China, Nov. 2019.

[25] B. Li and L. Han, "Distance weighted cosine similarity measure for text classification," in *Proceedings of the 14th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL)*, Hefei, China, Oct. 2013.

[26] *GDPR*, https://gdpr-info.eu/, [Accessed: 2024-04-11].

[27] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional LSTM networks for improved phoneme classification and recognition," in *Proceedings of the 15th International Conference on Artificial Neural Networks. (ICANN)*, Warsaw, Poland, Sep. 2005.

[28] *Lingua*, https://github.com/pemistahl/lingua-py, [Accessed: 2024-04-11].

[29] C. E. Herley, F. Tu, and J. Pillai, *Detecting and mitigating abusive network activity based on versioned browser usage*, US Patent App. 17/852,267, Dec. 2023.

[30] *Pytorch*, https://pytorch.org/, [Accessed: 2024-04-11].

[31] K. Song, X. Tan, T. Qin, J. Lu, and T. Liu, "MPNet: Masked and permuted pre-training for language understanding," in *Proceedings of the 34rd Conference on Neural Information Processing Systems (NeurIPS)*, Virtual Conference, Dec. 2020.

[32] *Sentence-transformers/all-mpnet-base-v2*, https://huggingface.co/sentence-transformers/all-mpnet-base-v2, [Accessed: 2024-04-11].

[33] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, New Orleans, LA, May 2019.

[34] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?" In *Proceedings of the 36rd Conference on Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, Nov. 2022.

[35] Y. Gong, D. Ran, X. He, T. Cong, A. Wang, and X. Wang, "Safety misalignment against large language models," in *Proceedings of the*

2025 Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, Feb. 2025.
[36] *Google chrome aims to solve account hijacking with device-bound cookies*, https://www.csoonline.com/article/2084025/google-chrome-aims-to-solve-account-hijacking-with-device-bound-cookies.html, [Accessed: 2024-04-11].
[37] M. Singh, "Privacy-preserving marketing analytics: Navigating the future of cookieless tracking," *International Journal of Enhanced Research in Management & Computer Applications*, vol. 13, pp. 2319–7471, Mar. 2024.
[38] A. G. Dobrakowski, A. Pacuk, P. Sankowski, M. Mucha, and P. Brach, "Improving ads-profitability using traffic-fingerprints," in *Proceedings of the 20th Australasian Data Mining Conference 2022 (AusDM)*, Western Sydney, Australia, Dec. 2022.
[39] J. Yang, S. Rahardja, and S. Rahardja, "Click fraud detection: HK-index for feature extraction from variable-length time series of user behavior," in *Proceedings of 32nd IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Xi'an, China, Aug. 2022.
[40] H. Xu, D. Liu, A. Koehl, H. Wang, and A. Stavrou, "Click fraud detection on the advertiser side," in *Proceedings of the 19th European Symposium on Research in Computer Security (ESORICS)*, Wroclaw, Poland, Sep. 2014.
[41] J. Kim, J. Park, and S. Son, "The abuser inside apps: Finding the culprit committing mobile ad fraud," in *Proceedings of the 2021 Annual Network and Distributed System Security Symposium (NDSS)*, Virtual Conference, Feb. 2021.
[42] C. Cao, Y. Gao, Y. Luo, M. Xia, W. Dong, C. Chen, and X. Liu, "AdSherlock: Efficient and deployable click fraud detection for mobile applications," *IEEE Trans. Mob. Comput.*, vol. 20, no. 4, pp. 1285–1297, Apr. 2021.
[43] F. Dong, H. Wang, L. Li, Y. Guo, T. F. Bissyandé, T. Liu, G. Xu, and J. Klein, "FraudDroid: Automated ad fraud detection for android apps," in *Proceedings of the 2018 ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*, Lake Buena Vista, FL, Nov. 2018.
[44] H. Lee, J. Lee, D. Kim, S. Jana, I. Shin, and S. Son, "AdCube: WebVR ad fraud and practical confinement of third-party ads," in *Proceedings of the 30th USENIX Security Symposium (Security)*, Virtual Conference, Aug. 2021.
[45] O. Alrawi, C. Lever, K. Valakuzhy, R. Court, K. Z. Snow, F. Monrose, and M. Antonakakis, "The circle of life: A large-scale study of the IoT malware lifecycle," in *Proceedings of the 30th USENIX Security Symposium (Security)*, Virtual Conference, Aug. 2021.
[46] M. Yao, J. Fuller, R. P. Kasturi, S. Agarwal, A. K. Sikder, and B. Saltaformaggio, "Hiding in plain sight: An empirical study of web application abuse in malware," in *Proceedings of the 32nd USENIX Security Symposium (Security)*, Anaheim, CA, Aug. 2023.
[47] J. Fuller, R. P. Kasturi, A. K. Sikder, H. Xu, B. Arik, V. Verma, E. Asdar, and B. Saltaformaggio, "C3PO: large-scale study of covert monitoring of c&c servers via over-permissioned protocol infiltration," in *Proceedings of the 28th ACM Conference on Computer and Communications Security (CCS)*, Seoul, South Korea, Nov. 2021.
[48] R. P. Kasturi, J. Fuller, Y. Sun, O. Chabklo, A. Rodriguez, J. Park, and B. Saltaformaggio, "Mistrust plugins you must: A large-scale study of malicious plugins in wordpress marketplaces," in *Proceedings of the 31st USENIX Security Symposium (Security)*, Boston, MA, Aug. 2022.
[49] O. Alrawi, M. Ike, M. Pruett, R. P. Kasturi, S. Barua, T. Hirani, B. Hill, and B. Saltaformaggio, "Forecasting malware capabilities from cyber attack memory images," in *Proceedings of the 30th USENIX Security Symposium (Security)*, Virtual Conference, Aug. 2021.
[50] J. Fuller, M. Yao, S. Agarwal, S. Barua, T. Hirani, A. K. Sikder, and B. Saltaformaggio, "Enhanced Web Application Security Through Proactive Dead Drop Resolver Remediation," in *Proceedings of the 32nd ACM Conference on Computer and Communications Security (CCS)*, Taipei, Taiwan, Oct. 2025.
[51] S. Wang, C. Liu, X. Gao, H. Qu, and W. Xu, "Session-based fraud detection in online e-commerce transactions using recurrent neural networks," in *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD)*, Skopje, Macedonia, Sep. 2017.
[52] B. Branco, P. Abreu, A. S. Gomes, M. S. C. Almeida, J. T. Ascensão, and P. Bizarro, "Interleaved sequence rnns for fraud detection," in *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, Virtual Conference, Aug. 2020.
[53] Z. Wang, Q. Wu, B. Zheng, J. Wang, K. Huang, and Y. Shi, "Sequence as genes: An user behavior modeling framework for fraud transaction detection in e-commerce," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, Long Beach, CA, Aug. 2023.
[54] C. Liu, L. Sun, X. Ao, J. Feng, Q. He, and H. Yang, "Intention-aware heterogeneous graph attention networks for fraud transactions detection," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, Virtual Conference, Aug. 2021.
[55] X. Li, W. Yu, T. Luwang, J. Zheng, X. Qiu, J. Zhao, L. Xia, and Y. Li, "Transaction fraud detection using GRU-centered sandwich-structured model," in *Proceedings of the 22nd IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, Nanjing, China, May 2018.
[56] M. Yao, R. Zhang, H. Xu, S. Chou, V. C. Paturi, A. K. Sikder, and B. Saltaformaggio, "Pulling off the mask: Forensic analysis of the deceptive creator wallets behind smart contract fraud," in *Proceedings of the 45th IEEE Symposium on Security and Privacy (S&P)*, San Francisco, CA, May 2024.
[57] J. Zhang, Y. Xie, F. Yu, D. Soukal, and W. Lee, "Intention and origination: An inside look at large-scale bot queries," in *Proceedings of the 20th Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, Feb. 2013.

Appendix A. Session Slicing Threshold Selection

Table 9 presents coverage and similarity statistics across different time-based thresholds for session generation, assessing their impact on intra-session and inter-session similarity. These statistics are derived from the traffic of 10,000 randomly sampled organic users. Column 1 lists the session-slicing thresholds, ranging from 10 to 120 minutes. Since CoSEC relies on session-level information for detecting ad-click fraud, a session is considered a valid input only if it contains at least three search requests after slicing. Consequently, consecutive search requests from the same user may form multiple sessions, some of which may not meet this criterion. We define a user's traffic (identified by Cookie ID) as covered by CoSEC if at least one session from that user qualifies as a valid input, as shown in Column 2. Additionally, Column 3 reports request-level coverage, representing the total number of search requests within these valid sessions across the entire dataset. To evaluate session coherence, we compute session query similarity using the method described in §3.2.1 and report its percentiles (5%, 25%, median, 75%, and 95%) along with the average in Columns 4-9. This metric quantifies intra-session similarity, capturing the semantic consistency of queries within a session. To assess inter-session continuity, we measure the query similarity between the last query of an earlier session and the first query of a later session when a user's traffic spans multiple sessions. Columns 10-15 present the percentiles and average of this

TABLE 9: Coverage and Similarity Statistics Over Different Time Intervals

Threshold	Coverage (%)		Intra-Session Similarity						Inter-Session Similarity					
	%Users	%Request	5%	25%	Median	75%	% 95%	Avg.	5%	25%	Median	75%	% 95%	Avg.
10 mins	86.4%	83.5%	0.106	0.295	0.452	0.629	0.908	0.471	0.018	0.104	0.203	0.365	0.920	0.275
20 mins	89.6%	87.2%	0.106	0.295	0.445	0.614	0.896	0.463	0.018	0.097	0.187	0.349	0.892	0.263
30 mins	90.8%	88.9%	0.108	0.296	0.441	0.606	0.890	0.460	0.015	0.094	0.179	0.343	0.893	0.258
40 mins	91.7%	90.1%	0.108	0.296	0.438	0.602	0.886	0.458	0.012	0.091	0.175	0.338	0.902	0.254
50 mins	92.3%	90.8%	0.108	0.297	0.436	0.596	0.887	0.457	0.012	0.090	0.175	0.338	0.921	0.254
60 mins	**92.6%**	**91.4%**	**0.108**	**0.296**	**0.434**	**0.594**	**0.884**	**0.455**	**0.011**	**0.090**	**0.172**	**0.336**	**0.972**	**0.254**
70 mins	92.9%	91.9%	0.108	0.296	0.433	0.592	0.882	0.454	0.011	0.089	0.172	0.336	0.947	0.254
80 mins	93.2%	92.2%	0.108	0.295	0.431	0.590	0.878	0.453	0.011	0.088	0.171	0.335	0.982	0.254
90 mins	93.4%	92.6%	0.109	0.295	0.430	0.589	0.877	0.452	0.009	0.088	0.169	0.335	1.000	0.254
100 mins	93.7%	92.8%	0.110	0.296	0.429	0.588	0.875	0.451	0.010	0.088	0.167	0.334	1.000	0.254
110 mins	93.9%	93.0%	0.110	0.296	0.428	0.587	0.873	0.451	0.009	0.087	0.167	0.335	1.000	0.254
120 mins	94.0%	93.3%	0.110	0.296	0.427	0.586	0.874	0.451	0.009	0.087	0.166	0.333	1.000	0.254

TABLE 10: FEATURE DESCRIPTIONS AND SCALED VALUES

Feature Name	Feature Description	Scaled Value
Session Length	Number of requests in the session	Total Count
Time Span	Total time duration of the session	Total Seconds
Time Interval	Time intervals between search requests in a session	Min, Max, Average
Page Number	Page number requested in the search results	Min, Max, Average
Clicks	Number of clicks made on search results	Min, Max, Average
Returned Ad Count	Number of ads returned in the search results	Min, Max, Average
Revenue	Revenue generated from the session	Min, Max, Average, Sum
Query Length*	Number of characters in the query string	Min, Max, Average
Query Word Count*	Number of words in the query string	Min, Max, Average
Query Similarity*	Similarity between successive queries in the session	Min, Max, Average

*: Query Similarity requires computing resource-intensive query embedding, resulting in an overhead similar to CoSeC.

TABLE 11: Overhead Measurement.

Dataset	Period	Embedding Phase			Inference Phase		
		Time (min)	Avg. Mem	Avg. GPU Mem	Time (min)	Avg. Mem	Avg. GPU Mem
Evaluation Dataset	Feb 1^{st} - 14^{th}	277.17	2.89 GB	0.75 GB	0.47	6.05 GB	0.72 GB
Testing Dataset 1	Mar $1^{st} - 7^{th}$	282.35	2.87 GB	0.76 GB	0.47	6.02 GB	0.72 GB
Testing Dataset 2	Mar $8^{th} - 14^{th}$	276.90	2.88 GB	0.77 GB	0.46	5.98 GB	0.71 GB
Testing Dataset 3	Mar $15^{th} - 21^{st}$	270.32	2.92 GB	0.74 GB	0.47	6.12 GB	0.72 GB
Testing Dataset 4	Mar $22^{nd} - 28^{th}$	288.42	2.89 GB	0.75 GB	0.47	6.00 GB	0.72 GB
Total	N/A	279.3	2.89 GB	0.75 GB	0.47	6.03 GB	0.72 GB

inter-session similarity distribution, reflecting the degree of semantic separation between adjacent sessions.

For threshold selection, a larger threshold T results in more individual requests from the same user being grouped into a single session. This leads to longer sessions and higher coverage at both the user and request levels, as shown in Columns 2 and 3. Notably, as session length increases, the diversity of search topics within a session also grows, causing intra-session similarity to decrease. In addition, with a longer enforced temporal gap between sessions, we assume that users are more likely to shift search topics, leading to lower inter-session similarity.

Our goal is to identify a proof-of-concept threshold for session generation that balances high coverage and intra-session similarity with relatively low inter-session similarity. Based on this, we select 60 minutes as the session threshold in our experiments. First, coverage stabilizes beyond this point, with only marginal gains at higher thresholds (e.g., 92.6% of sessions and 91.4% of requests at 60 minutes, compared to 94.0% and 93.3% at 120 minutes). Second, intra-session similarity remains high, with a median of 0.434 and an average of 0.455, preserving meaningful behavioral patterns within sessions. In contrast, inter-session similarity is sufficiently low, with a median of 0.172 and an average of 0.254, ensuring that sessions remain distinct in terms of search topics. This selection effectively balances coverage, coherence within sessions, and topic differentiation between sessions, making 60 minutes a suitable threshold for our framework.

Appendix B. Statistical Features For SOTA Baseline Models

Table 10 shows the statistical features used in §5.4 for baseline models. We collected these statistical features from the same raw session knowledge available to CoSeC, which is assumed to be available to most search engines. The first two rows of Table 10 show our collecting the

session length in the count of requests and the overall period in seconds. Rows 3-9 show our collecting the minimum, maximum, and average values for time intervals, query lengths, query word counts, query similarities, page numbers, clicks, returned ad counts, and the ad revenue from requests in a session. We additionally add the total revenue to the statistical feature set. Notably, as introduced in §3.2, getting query similarity features requires embedding queries with CoSeC's language model in advance. Since this process accounts for the major overhead for CoSeC, as shown in Table 11, models depending on these statistical features will cause a similar overhead to CoSeC.

Appendix C. Overhead Measurement

Search engines may receive millions or even billions of search sessions daily. A deployed fraud detection algorithm should incur minimal cost while capturing significant fraudulent traffic with high precision. To ensure CoSeC's prototype meets this requirement, we measure the overhead of deploying CoSeC for prediction. We tested the execution overhead on our testbed introduced in §5.1 using 100,000 sessions from each dataset. We used a single-thread Python program to embed features from each search request and a batch size of 64 for fraud prediction. Table 11 lists the execution time and resource breakdown.

On average, the program processes embedding for 100,000 sessions in 279.3 minutes, with an average memory usage of 2.89 GB and 0.75 GB GPU memory usage during embedding. For prediction, CoSeC processes 100,000 sessions within 0.47 minutes, with memory usage of 6.03 GB and 0.72 GB GPU memory usage. Given that this overhead execution is done on a bare-metal testbench,CoSeC can be easily deployed in a more robust and faster production environment with reasonable resource costs.

Appendix D. Top Search Engines' Search Suggestion

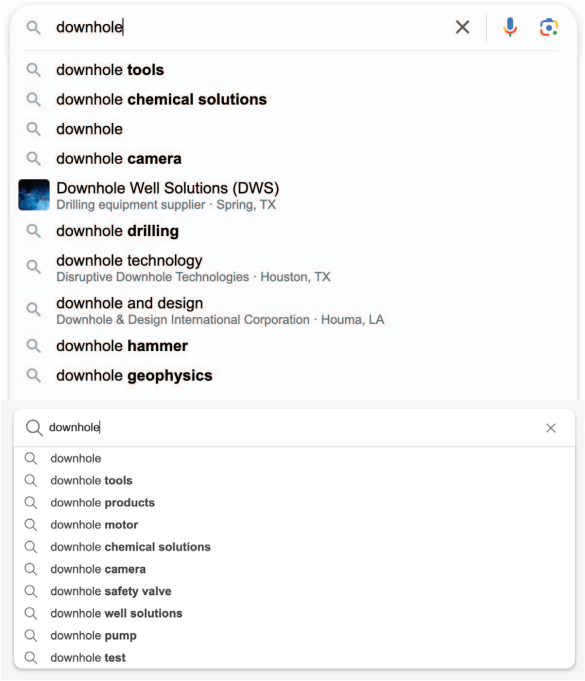

Figure 9: Top Search Engine's Searching Suggestion Results From Typed Keyword.

Appendix E.
Meta-Review

The following meta-review was prepared by the program committee for the 2025 IEEE Symposium on Security and Privacy (S&P) as part of the review process as detailed in the call for papers.

E.1. Summary

The paper investigates the ability of linguistic similarity between successive queries in a search query sequence to identify search click fraud. The paper shows that statistical features extracted from search sessions are ineffective since they exhibit overlap between fraudulent and organic sessions. Instead, the paper combines such features with cosine similarity of successive queries, and also temporal features, to develop a COSEC model that achieves improved precision and recall over state-of-the-art search click fraud detection solutions. The paper further discusses case studies that show how fraudsters evolve their strategies, including using search suggestions to increase the similarity of their query sessions.

E.2. Scientific Contributions

- Provides a Valuable Step Forward in an Established Field

E.3. Reasons for Acceptance

1) The paper provides a valuable step forward in an established field. Search click fraud is a well-known and non-stationary problem. COSEC takes a principled statistical approach to evaluating session coherence and identifying fraudulent sessions. COSEC is also evaluated via case studies in a real-world search ad context, demonstrating its practicality.

E.4. Noteworthy Concerns

1) The work does not evaluate how an adaptive adversary could thwart COSEC's protections by reverse engineering COSEC's methods and model.
2) The dataset used for evaluation does not span the entire space of search behavior but rather contains two classes of known-good and known-fraudulent search activity. As such, the evaluated performance of COSEC may not generalize to the full population of search sessions. However, acquiring such a dataset was deemed not technically feasible by the reviewers, so the dataset used in the work is considered a substantial contribution that will enable defenses and future science.