It's All in the Name: Why Some URLs are More Vulnerable to Typosquatting

***Abstract*—Typosquatting is a blackhat practice that relies on human error and low-cost domain registrations to hijack legitimate traffic from well-established websites. The technique is typically used for phishing, driving traffic towards competitors or disseminating indecent or malicious content and as such remains a concern for businesses.**

We take a fresh new look at this well-studied phenomenon to explore why some URLs are more vulnerable to typing mistakes than others. We explore the relationship between human hand anatomy, keyboard layouts and typing mistakes using various URL datasets. We create an extensive user-centric typographical model and compute a Hardness-Quotient (likelihood of mistyping) for each URL using a quantitative measure for finger and hand effort. Furthermore, our model predicts the most likely typos for each URL which can then be defensively registered. Cross-validation against actual URL and DNS datasets suggests that this is a meaningful and effective defense mechanism.

I. Introduction

Authentication is perhaps the cornerstone of network security; all preventive measures are nullified if an entity cannot successfully confirm who they are communicating with. The first step towards authentication on the web, even before DNS lookups or HTTPS connections, is the action of a user typing the domain name of a site they wish to visit. Typosquatting attacks the authentication process at this moment: the exploit happens before any of these other mechanisms from the "security stack" come into play.

Typosquatting relies on human error and low-cost domain registrations in order to hijack legitimate traffic away from well established websites[1]. An attacker can register domain names similar to well known websites in the hope that a naive typographic mistake from a user will direct him to a typo domain rather than the intended website [2], [3]. Typosquatters use a variety of techniques to profit directly or indirectly from such hijacked traffic. One of the most common ways to make money is through pay-per-click or affiliate advertising. Phishing attempts and malware infections using drive-by downloads and malicious ads have also been observed as a consequence of such typographical mistakes [4]. It is important to note that the user may not even realize that they are visiting a potentially malicious website. Competitor-squatting is an interesting subphenomenon where businesses can ride on the coattails of their successful competitors by registering mistyped domain names and redirecting traffic towards themselves[1]. Since domain registrations are relatively inexpensive, typosquatted websites can cause damage to business and brand equity with little or no investment.

While protection mechanisms for typosquatting have mostly focused on defensive registrations [5], [4], in this paper, we make the case that a better understanding of why we mistype URLs in the first place can enable more effective defensive strategies. For instance, the difficulty of typing a specific domain name, or likelihood of making a certain type of mistake, is not a purely random process. In other words, some typos are more likely to happen than others. Hence, the very first defense one can enact is to choose a domain name that is less likely to be mistyped. Furthermore, having an understanding of which typos are more likely than others can lead to smarter defensive registrations. In fact, Khan et al. found that many of the losses surrounding typos are simply due to the (likely inevitable) action of mistyping, rather than the efforts of typosquatters [6]. Improving the likelihood of a user typing a domain name correctly can thus improve both the authenticity and availability of an Internet domain.

We take a user-centric approach to typosquatting in which we study the relationship between human hand anatomy, keyboard layouts and morphological features of the URLs being typed in order to calculate the probability of a typing error. Due to certain anatomical factors, such as the structure and muscular make-up of the index finger as opposed to the little finger, and the way in which a standard QWERTY keyboard is laid out, our study finds that certain combinations of letters and characters are more susceptible to typing mistakes than others. This leads to a systematic pattern in the mistyped URLs. We argue that these error patterns can be used to predict the most likely set of mistyped URLs for any base URL. We argue that defensively registering this potential *squat space* can be an effective strategy against typosquatting. To the best of our knowledge, there has been no previous research that adopts a user-centric approach along with an anatomical typographical model to analyze the phenomenon of typosquatting.

To summarize, we make the following contributions in our work:

Dataset Collection: We conduct an extensive fixed-text keystroke study targeting the speed and accuracy of typing text and domain names to develop a clear understanding of the patterns and intricacies involved in URL-specific typing. We also gathered longitudinal domain name data for roughly 10 million typed URLs (including mistyped ones) from an academic institution over a period of 4 months. In addition, we augment our collected data with 3 large corpora of URLs typed by real-world users acquired from URLFixer (a popular browser plugin) [7], BlueCoad Web Proxy [8] and a dataset from Indiana University [9].

Typographical Model: Using the aforementioned datasets, we explore the various anatomical and keyboard-layout features that make certain URLs more vulnerable to mistakes than others. Having established that certain combinations of letters and characters are more prone to errors, we present a quantitative typographical machine learning model for measuring the

difficulty of typing different URLs (Hardness Quotient). The Hardness Quotient is an absolute metric and can be used by businesses to compare candidate domain names so as to choose the one that is less prone to typing mistakes and consequently less vulnerable to typosquatting.

Prediction Service: Based on the typographical model, we develop a prediction service, that allows us to identify the most likely domains that a brand needs to register to minimize traffic hijacking via typosquatting. Since all such domain names might not be feasible to purchase and manage, the service outputs a ranked list of the most relevant ones that cover a large chunk of the squat space. As an application of the prediction service, we developed a browser plugin, which detects a typo and predicts the most likely alternatives using our system.

The rest of the paper is organized as follows: Section II covers background and related work on typosquatting. Section III describes our datasets and how we processed and cleaned them. Section IV presents the typographical model. Using the model we build a prediction service in Section V and show our main results in Section VI. Before we conclude in Section VIII, we present a few limitations and discuss some additional aspects of our work in Section VII.

II. Related Work

Typosquatting has its origins in "cyber-squatting" where money could be made by registering domains that could eventually be sold off to the rightful owners at a higher premium [10], [11]. Typosquatting is a natural extension of the idea that relies on human error and low-cost domain registrations in order to hijack legitimate traffic from well established websites [1], [4], [12]. Domain names similar to well known websites are registered in the hope that a naive typographic mistake from a user will direct him to a typo domain rather than the intended website [2], [3]. Edelman' et al. [1] were the first to systematically study the phenomenon. Subsequent related work mostly falls into one of two categories: work meant to quantify user harm and work aimed at understanding and detecting Typosquatting activities.

Understanding how typosquatters choose which domain name typos to target is the most important step in combating this issue. Virtually all countermeasures, defensive registrations as well as post-registration lawsuits [13], heavily depend on knowing which URLs are chosen for typosquatting. To this end, Mohaisen et al. [4] provide a comprehensive overview of previous studies which focus on various paths that typosquatters take in choosing domain names to register. Studies which try to predict the domain names that typosquatters target either focus on specialized cases of typos, for example homophonic typos and bit flipping typos, or on lexically similar typos which are made by real typing errors [4]. Since we analyze the relationship between two typed URLs from a lexical perspective, we focus on the works which studied this area of typosquatting, as opposed to those which concentrated on specialized mistakes.

Typical approaches to generate the typosquatting domain of the lexically similar counterparts of popular websites, concentrate on generating all domain names which fall under a certain lexical metric [4], [14]. One such measure of string similarity is the the *Damerau-Levenshtein* distance metric which measures the number of insertions, deletions, permutations and substitutions needed to transform one string into another [15], [16].

On a parallel stream, the value-chain and side-effects of typosquatting have also been studied extensively. A typosquatter can use a variety of techniques to profit directly or indirectly from such domains. One of the most common ways to make money is through pay-per-click or affiliate advertising. Phishing attempts and malware infections using drive-by downloads and malicious ads have also been observed as a consequences of such typographical mistakes [4]. Moore et al. [1] studied the monetary incentives for typosquatting. One of their key findings was that pay-per-click advertising and affiliate marketing were the most important sources of revenue. Khan et al. [6] try to estimate the lost time to end-users experiencing typosquatting. [14] demonstrated that despite increased awareness and defensive registrations, typosquatting is not restricted to only popular websites and has a long-tail effecting less popular websites as well. Typosquatted websites have also been used as propagation vectors for malware and phishing attempts. "Competitor typosquatting" is an interesting sub-phenomenon where businesses can ride on the coat-tails of their successful competitors by registering mistyped domain names and redirecting traffic towards themselves [1]. This redirection becomes even more dangerous given that a simple user may not realize that they are visiting a counterfeit or potentially malicious website and may give out sensitive information.

To the best of our knowledge, ours is the first study that takes a human-centered view of the typosquatting problem. We study the relationship between human hand anatomy, keyboard layouts and real-world typing mistakes in order to explain why certain URLs are more vulnerable to typing errors given that human beings tend to conserve energy and minimize effort [17], [18]. In the process, we have come up with a typing difficulty metric for each word based on a standard QWERTY keyboard layout. The closest work to our model is Carpalx [19] that measures typing effort to produce an optimized keyboard layout which minimizes carpal strain during typing, a goal orthogonal to ours. Others in the keyboard layout community have also attempted to reduce typing effort by proposing new layouts [20], [21]. Our scheme, leverages some of their insights however, since our model is built on findings from actual data some metrics and scores are quite different from those of the keyboard layout community and have been borrowed from medical science studies.

III. Datasets Acquisition and Processing

In our study, we mainly relied on two large datasets that we collected ourselves. However, we also used numerous other datasets that we acquired from other sources to improve the

accuracy of the classifier. We explain each dataset we used below:

Fixed-Text Keystroke Study (DS-1): To the best of our knowledge, we conducted the first of a kind keystroke study focused primarily on understanding typing errors. The entire exercise was designed to highlight where and when users make typing errors while typing URLs and allowed us to hypothesize as to why human beings make particular typing errors. To this end, we deployed a website which presented users with a range of text passages which they had to type word-for-word in a text box below. We logged every single key pressed by the user and were able to extract the entire range of errors (even those which the users themselves corrected using the backspace or delete keys). The subjects of the study were asked to take 4 tests with each test falling into one of three main categories: URL names, words which are typed entirely by either the left or right hand, and fixed text typed using both hands. Each test was designed to explore a wide range of typing observations (such as differences between the typing proficiency in the left and the right hand, biases towards straightening of the fingers as opposed to curling them inwards etc.). These sample texts presented a spectrum of difficult-to-type patterns to the user so that we could see a diverse set of errors resulting from a large range of features that we engineered into the tests. For instance, repeated keystrokes with the pinky finger (e.g., *zaqzaqzaq.com*, which is an actual domain name) or the index finger (e.g., *junjun.com*, again an actual domain name) or row changes from top to bottom and back (e.g., *mimub.com*, again a real domain) etc. These domain names were extracted from the .com and .net zone files. The participants were all students from the computer science department at a local university, which meant they were proficient at typing with both hands. A group of 59 people participated in our study. Together the tests consisted of 600 unique words out of which 134 were mistyped and 466 did not exhibit any typos, which backs up the claim that certain strings are easier to type on certain keyboards. The error breakdown is shown in Table I.

Error Type	DS-1	DS-2	DS-3	DS-4	DS-5
Single Character Omission	21.43	15.08	21.4	46.32	39.37
Double Character Omission	3.57	2.66	2.89	4.08	2.68
Single Character Insertion	14.29	36.14	38.23	23.5	24.19
Double Character Insertion	3.57	18.18	5.1	0.0	0.06
Single Character Replacement	39.29	16.19	16.45	20.68	24.98
Double Character Replacement	3.57	0.22	0.76	0.0	0.0
Character Swap	7.14	2.44	4.19	2.29	4.82
Character Repeat	7.14	9.09	10.97	3.18	3.89

TABLE I
ERROR BREAKDOWN OF ALL DATASETS THAT WE USED. VALUES SHOWN ARE ALL IN PERCENTAGES.

Hostel URL Dataset (DS-2): We installed a firewall in a university to monitor the URLs accessed by the students residing in the hostels. Before collecting any data from the firewall, we received a formal approval from our local Institutional Review Board (IRB) because our data included detailed browsing activity of the users. We implemented several mechanisms to protect user privacy. For example, we removed any personally identifiable information from the URLs and only stored the domain names. Since we were only passively collecting data, no normal user activity was impacted by our experiments. Each time a URL was typed in a browser, the firewall received the request first and allowed us to log the domain name that was being requested by the user. We collected this data over a period of 4 months for approximately 2000 students. The total number of URLs typed was around 8 million, 2.7% of which were typos (breakdown in Table I).

URL Fixer Dataset (DS-3): The third dataset [22] consisted of the URLs entered by real-world users of the URL Fixer browser add-on[7]. URL Fixer detects a mistyped TLD domain in the URL typed and alerts a user to the mistake in real-time (For example, if you type google.con, it corrects it to google.com). This dataset included roughly 12,000 URLs, both correct and mistyped, over a 9 month period between February and October 2011. Approximately 2% of these URLs were typos and this dataset was primarily used for cross validating our model.

Bluecoat Dataset (DS-4): This dataset contains standard web proxy logs generated via the BlueCoat web prox [8] for 2005. The proxy is deployed in a lab network. The dataset contains roughly 1.8 million URLs out of which 38,000 (2.14%) are typos.

Indiana University Click Dataset (DS-5): This dataset consists of HTTP requests made from Indiana University [9] for 2009. The dataset was collected using the Berkeley Packet Filter and then regular expressions were used to extract HTTP GET requests. Total number of URLs in this dataset are roughly 30 million out of which 400,000 (1.4%) are typographical errors.

We now explain how we processed and cleaned the datasets and extracted human typed URLs from each corpus.

1) Extracting Hand-Typed URLs: We first outline our mechanisms for identifying typed URLs as opposed to auto-generated/redirected ones in our datasets. The firewall logs, proxy logs and HTTP requests included URLs that were auto-generated, most probably through an application's URL requests and URL redirections. These URLs were likely to skew our model and alter the results of our prediction service. Hence, the datasets needed to be cleaned so as to contain only the hand typed domain names. To solve this issue, we clustered our dataset using seeded centroids. In the first phase, we extracted all members of Alexa's top 10k URLs (as these are frequently targeted by typosquatters) from the datasets and then used these base URLs as centroids around which the remaining URLs were clustered (the exact technique is explained in the next subsection). This resulted in an efficient and simple method to rule out domain names, which were not hand typed as only those strings that could realistically be mistyped versions of the centroid were added to a cluster. Furthermore, this strategy also allowed us to prune the dataset by removing outliers and focus on the more popular URLs that are commonly targeted by attackers. In parallel, we also applied simple heuristics, such as URL size, to further filter out the hand typed URLs as extremely large instances were highly unlikely to be hand typed (for example p4-f5wxpcrz2dp5k-

r5kvtsjmgsz7e4du-875034-i1-v6exp3-ds.metric.gstatic.com).

2) Identifying Mistyped URLs: Another challenge was to identify if a given URL is a mistyped version of another URL or not. Our method of categorizing URLs was relatively simple and a hybrid of three known distances. We combined the Levenshtein distance [16] with the keyboard key distance (which allowed us to rule out unrealistic candidate URLs, such as zoogle instead of google because of the penalty imposed due to the distance between 'z' and 'g'). The basic steps to categorize a candidate URL as a typo are as follows: first, find the hybrid edit distance between the two URLs. Second, compare the resulting hybrid distance with 20% of the length of the shorter URL. If the hybrid distance is less than the 20% value the candidate URL is marked as a typo. The 20% value was experimentally derived as it gave the best results with the least false positives. To illustrate this better consider the following; in a vanilla string edit distance calculation, the difference between *google* and *foogle* would be the same as the distance between *google* and *woogle*; a distance of 1. However, in the hybrid distance algorithm, using the key distance as the penalty, the distance between *google* and *foogle* would be 1, whereas the distance between *google* and *woogle* would be 4. This is a more realistic representation of how likely a person is to mistype *foogle* in place of *google* as opposed to *woogle*. Once the algorithm finishes execution, we are left with clusters around a base URL, each member of which is a mistyped version of the base centroid of the cluster. At this stage we apply a modified version of the Ratcliff-Obershelp algorithm [23], which is commonly used for approximate string matching, to further prune the results (only URLs with more than 80% match with the centroid are kept and others are discarded).

IV. Typographical Model

In order to determine if a URL is more prone to typographic mistakes than others, we need a precise typographic model that captures the relationship of underlying keyboard layouts, hand anatomy/morphology and the lexical properties of the URL itself. To this end, we **gather domain knowledge** by collecting material relevant to the model e.g., work done by the keyboard layout and medical communities. Additionally, we also explored research pertaining to the lexical properties of a string using n-gram based analysis. Then we **extract insights** by indentifying metrics that contribute to typing errors using gathered domain knowledge. We then move on to **feature engineering** by extracting meaningful features from the set of insights gathered in the previous phase. Finally we do **weight evaluation** of our features through a classifier, thus ranking them in order of importance.

To test the accuracy of our model we perform cross-validation against a held-out sample from our datasets (DS-3 and part of DS-5). We demonstrate in Section VI that on average, words and URLs that have a higher hardness quotient also have higher corresponding mistyped instances showing a strong correlation. We also build a prediction service to aid in defensive registration based on our model and test the service on actual data we collected to further corroborate the accuracy of our model.

A. Domain Knowledge

A lot of research has been conducted in the past on different aspects of typing ranging from anatomy of hands and posture to different typing habits. We have found some interesting insights into the reasons for typing errors. In the interest of space, we discuss each insight very briefly and omit some of the other insights extracted from the gathered pool of domain knowledge.

1) Typing & Hand Anatomy: The tendons of muscles which flex the fingers pass through a common sheath in the carpal canal. Researchers have used the movement of these tendons as an indicator of biomechanical stress [24]. This is because continuous sliding of tendons over one another during extremely repetitive task such as typing leads to friction [25], which can cause problems in the tendons and adjacent nerves [24], such as carpal tunnel syndrome and tendinitis [26].Tendon travel, a measure of musculoskeletal discomfort [27], is most when the wrist is continuously extended outwards and fingers joints bent at large angles [27], as would be the case in typing. **Insight 1: Switching fingers vigorously while typing URLs increases tendon travel causing discomfort and fatigue which can result in typing errors.**

Carpal Tunnel Pressure (CTP), another measure of discomfort in hands[28], shows a massive increase with constant wrist extension and flexion or radial (inwards) and ulnar (outwards) deviation [29]. This increase is more pronounced when the wrists are initially positioned in a tilt (forearm and wrist are not aligned) [30], as is typically the case in typing [31]. Studies suggest that CTP can be lowered by avoiding full finger flexions [32], which are normal in typing, especially the lower row. One study also finds that with extended wrists, extended or straight fingers elevate CTP [32], which implies that the upper row can be a cause of discomfort as well.**Insight 2: Upper and bottom row typing is more strenuous for humans and hence more prone to errors.**

Ulnar deviations or outward bending of the wrist e.g. to reach far-off keys via the little fingers causes substantial discomfort [33]. **Insight 3: typing with the little finger to reach keys on the edges of the keyboard may lead to errors.**

2) Cognitive Control of Typing: Fingers on the same hand are "partnered" meaning that if one finger is extended or flexed the adjacent fingers exhibit similar tendencies, to some degree [34]. The user might not actively press with the incorrect finger along with the correct one, the aforementioned interrelation might generate some movement in the non-typing fingers, such as leaving them out of position [35]. This is known as the *enslavement effect* [36]. **Insight 4: Consecutive letters on adjacent fingers might be more prone to error since adjacent fingers are moved out of position.**

It was discovered that correlated motor units of the hands were simultaneously excited due to neural input [37]. This "co-activation" results in adjacent fingers also exerting force

along with the actual finger. Researchers have found that co-activation was most prominent in the ring finger and progressively decreased when little finger, middle finger, index finger and thumb exerted the force [38].**Insight 5: Since fingers vary in independence of movement, some fingers might be more prone to errors than others. Specifically, the ring finger is more error prone, followed by the little and middle fingers.**

Typing is a parallelized task whereby the brain is already planning to type several character in advance of the character being typed [35]. This parallel processing leads to "overlapped" movements where the brain plans in advance and overlaps movements of different fingers and hands [35] i.e., while a character is being typed, the other fingers and hand might be moving to get in position to type other characters. Overlapped movements can occur both within and across hand movements [35]. Researchers have argued that parallel processing and overlapping of movements result in incorrect finger assignment e.g., pressing with index finger of right hand instead of index finger of left hand [39].**Insight 6: If pre-planing and overlapping results in complex patterns and extensive movements of the hands and fingers, the brain might get confused resulting in incorrect finger or hand assignments.**

On the contrary, studies have also found that successive letters on alternate hands are at times faster to type [40]. **Insight 7: Typing successive characters through alternate hands has a noticeable effect on typing: it might make a URL more error prone or more error proof hence, this characteristic of a URL should definitely be explored.**

3) Text Properties: Researchers found that words belonging to the English language, more importantly words in routine use, were typed faster than random strings [35]because the brain already knows the combination of movements required to type a known word and it reduces the chances of error. **Insight 8: URLs comprising English words might be less error prone as compared to URLs made up of non-English words.**

As discussed earlier, typing is a parallel process but typing double letters or successive letters using the same finger is sequential and thus slower. Hence, it is possible that without waiting for the letter to be typed twice, another finger accidentally types another letter in between those double letters [35].Another common error is doubling of the wrong alphabet e.g., bokk instead of book [41]**Insight 9: Presence of a pair of alphabets in a URL might make it more error prone especially if the preceding or subsequent alphabets are pressed with a different finger or hand.**

Another error studied by researchers is the alternation reversal error where an incorrect letter is alternated on both sides of another letter in the middle e.g., 'were' may become 'wrer' or 'here' might become 'hrer' [42]. **Insight 10: URLs which contain same character on either side of a different character might be more error prone and exhibit a particular type of error.**

4) Keyboard Properties: It must be noted that QWERTY keyboard was originally designed to reduce the jamming of typebars [43] and is not an ergonomic design.It is estimated that 57% of keystrokes are allocated to the weaker left hand and thus it makes the design more prone to errors [44]. Other studies show that the left hand types slightly better [45]. **Insight 11: Hand proportion or division needs to be incorporated into the model and its effects on error frequencies should be studied further.**

Among other issues with the QWERTY layout, it pointed out that certain fingers are overworked while others are not assigned enough work (poor finger distribution). Home row has too little typing allocation and fingers have to jump from one row to another excessively causing fatigue [44]. Given how much the Dvorak design is discussed in literature, it is reasonable to treat these findings as insights and test them in our study. **Insight 12: URLs concentrated around a few fingers overwork the fingers. Insight 13: URLs which have too little typing on the home row may result in more errors. Insight 14: URLs which force the user to jump excessively between rows cause fatigue.**

B. Feature Engineering and Model Construction

1) Features: Features generated from the above discussed insights are described in Table II. We use these features to model the Hardness Quotient of URLs and then predict most likely typos of a given URL. Last two columns in Table II rank each feature in order of importance for calculating the Hardness Quotient and Prediction, along with the weights our model assigned to them.

2) N-grams: An n-gram is defined as a sliding sequence of *n* consecutive characters. For instance, for *facebook* and *n*=4, n-grams will be face, aceb, cebo, eboo and book.

3) Dictionary Generation: For any given URL and corresponding typos in our datasets, we generate an n-gram based dictionary. As an example, lets say we have *facebook* and its typo *facebok*. First, we align them using the Needleman-Wunsch algorithm [47] giving us *facebook* and *facebo-k*. Let's suppose the total occurrences of *facebook* were 1000, whereas occurrences of the typo *facebok* were 10. On n-gram level, the typo only resides in *eboo* and *book*, because all other n-grams are correct in the typo version as well. Hence, we will only populate our dictionary with the n-grams that have typos associated with them. These are stored as key-value pairs, keys being the n-grams and values being their *rate* of occurrence. For *book*, the rate would be 10/1000 = 0.01. Resultantly, our key-value pair will have key=*book* and value=*0.01*. If some other URL also contains this particular n-gram, e.g., *mybook.com* and it is mistyped as *myboik.com*, again the n-gram *book* has the typo. Let's say in this case *mybook* occured 500 times and *myboik* occured 10 times. Rate of error for *book* in this case will be 0.02. We will update this in our dictionary by adding the error rates, so in our dictionary, for the key of *book*, the value will be 0.01 + 0.02 = 0.03. We will do this for all n-grams which have typos associated with them in our dataset.

4) Training Phase: We use the Random Forest Regressor for our model. We train it by giving it a feature set and

Sr.	Feature Name	Description	Insight Used	(Rank) HQ Weight	(Rank) Prediction Weight
1	Finger Switches (FS)	No. of finger switches	1	(11) 0.033	(11) 0.026
2	Same Finger Count (SFC)	No. of successive characters on the same finger	1	(23) 0.015	(25) 0.010
3	Length	Length of URL	1, others	(24) 0.015	(2) 0.113
4	Distance	Total inter-keystroke distance	1	(5) 0.068	(5) 0.077
5	Top Row Streak (TRS)	Max no. of characters typed in the top row in one go	1, others	(16) 0.029	(15) 0.016
6	Home Row Streak (HRS)	Max no. of characters typed in the home row in one go	1, others	(30) 0.001	(28) 0.004
7	Bottom Row Streak (BRS)	Max no. of characters typed in the bottom row in one go	1, others	(32) 0.000	(31) 0.001
8	Numeral Row Streak (NRS)	Max no. of characters typed in the numeral row in one go	1, others	(31) 0.000	(32) 0.001
9	Sequence Break (SB)	SB occurs when a trigram fails to form a same row monotonic sequence	others	(15) 0.031	(7) 0.048
10	Cartesian Distance (CD)	CD from the source (‘G ’ and ‘H ’)	2	(4) 0.070	(4) 0.077
11	Top Row Count (TRC)	No. of characters in the upper row	2	(12) 0.032	(16) 0.014
12	Bottom Row Count (BRC)	No. of characters in the lower row	2	(13) 0.032	(27) 0.006
13	Home Row Count (HRC)	No. of characters on home row	2	(9) 0.038	(12) 0.023
14	Numeral Row Count (NRC)	No. of numerals above the upper row	2	(27) 0.004	(22) 0.012
15	Weak Finger Count (FG)	No. of characters on the edges of the keyboard typed by pinky finger	3	(7) 0.042	(10) 0.028
16	Adjacent Finger Count (AFC)	No. of successive characters on adjacent fingers	4	(10) 0.037	(19) 0.013
17	Total Base Hardness (TBH)	A penalty between 1-5 is assigned to each key representing its base difficulty. TBH is the sum of all the penalties for a particular URL (mapping available here [46])	5, others	(2) 0.079	(3) 0.097
18	Asymmetric Hand Shifting (AHS):	AHS occurs when two consecutive alphabets are typed using different fingers of different hands	6	(14) 0.032	(6) 0.055
19	Symmetric Hand Shifting (SHS)	SHS occurs when two consecutive alphabets are typed using same fingers of different hands	6	(17) 0.027	(18) 0.013
20	Hand Shifts (HS)	No. of successive characters on alternate hands	7	(18) 0.026	(24) 0.010
21	English Words (EW)	URLs consisting of English words (or substrings)	8	(26) 0.010	(21) 0.012
22	Same Finger Doubles (SFD)	No. of instances of two successive characters to be typed by the same finger	9	(22) 0.016	(23) 0.010
23	Double Characters (DC)	No. of double characters in a URL e.g. Google has 1 instance of double letters	9	(25) 0.011	(26) 0.007
24	Alternating Characters (AC)	No. of instances of same character on the either side of a different character e.g., the sequence *ere* in ‘there’	10	(21) 0.017	(17) 0.014
25	Left Hand Count (LHC)	No. of characters on left hand	11	(3) 0.073	(1) 0.120
26	Right Hand Streak (RHS)	Maximum number of keystrokes pressed using the right hand	11	(19) 0.026	(20) 0.012
27	Left Hand Streak (LHS)	Maximum number of keystrokes pressed using the left hand	11	(20) 0.022	(13) 0.020
28	Finger Count (FC)	No. of fingers involved in typing a URL	12	(8) 0.040	(14) 0.017
29	Home Row Distance (HRD)	Sum of the distance of all keys from the nearest home row key	13	(1) 0.085	(8) 0.039
30	Row Shifting (RS)	RS happens when two successive alphabets lie on different keyboard rows. A double shift is counted when the alphabets are spaced apart by two rows.	14	(6) 0.049	(9) 0.031
31	Upward/Downward Right Hand Progression (RHP)	RHP occurs when a trigram of three consecutive alphabets, one from each row, requires the right hand to type.	14	(28) 0.002	(30) 0.002
32	Upward/Downward Left Hand Progression (LHP)	LHP occurs when a trigram of three consecutive alphabets, one from each row, requires the left hand to type.	14	(29) 0.002	(29) 0.003

TABLE II
LIST OF ALL ENGINEERED METRICS USED IN THE MODEL ALONG WITH THEIR EXPLANATION, THE INSIGHT THAT HELPED ENGINEER THE METRIC AND THEIR CORRESPONDING WEIGHTS AND RANKS FOR THE TWO CLASSIFIERS USED IN THIS STUDY.

corresponding labels for all URLs in our training dataset. For each URL, we first calculate its features as discussed in Table II. This gives us a feature vector of length 32. We also create n-grams of the URL e.g., as discussed above. We will see if these n-grams are present in our dictionary and extract the corresponding error rate values. The rates are appended to the end of the feature vector giving us a vector of length 45 (if less we append zeros at the end). As an example, a vector might look like the following:

[Vector of 32 features, 0.04, 0.03, ..., 0.05, 0, 0, ..., 0]

The label for this vector will be the error rate of the URL e.g., for *facebook*, if it occured a 1000 times and was mistyped 240 times in total, our label will be 0.24. We will create feature vectors and compute their labels and feed them to our model for training.

5) Testing Phase: In order to test a URL, we will create its feature vector as described in the training phase. For instance, if we want to find the hardness quotient of *google.com*, we will calculate the 32 features of *google*, as mentioned in Table II. We will also compute the corresponding n-grams of *google* and extract their error rates from our dictionary. We will append these error rates to the feature vector giving us our final test vector of size 45. We will then input this to our trained Random Forest Regressor which will output a hardness quotient or probability of error for the URL, which is a value between 0 and 1.

V. PREDICTION SERVICE

The final piece of the puzzle is the service that uses our typographical model to predict the most likely typos for each URL that can be selectively registered. The prediction service allows existing domain owners to protect themselves from typosquatting by buying top k URLs from the *squat space*.

Squat space refers to the set of all possible URLs which are similar to the base URLand can be generated by different measures and criteria.We generate our squat space using a set of 14 possible error types e.g.,character repeat, character swap, character insertion, etc. as pointed out in previous work [5],

[1]. Depending on its features, the squat space of a URL can be quite huge e.g., a longer URL will have a larger squat space.

Smart Defensive Registrations: Defensive registrations refer to a situation where a URL owner buys URLs similar to the base URL in order to thwart attempts of typosquatting. Theoretically speaking, all URLs in the squat space can be typosquatted but it is impractical to buy and manage all those URLs, given the huge size of the squat space.Thus, the primary goal of the prediction service is to allow for smart defensive registration of a minimal URL set that is highly similar to the base URL and thus has higher probability of being typosquatted.

Prediction Scheme: The prediction mechanism incorporates the following analysis:

1) N-gram Dictionary Generation: Similar to the HQ dictionary, the concept of the n-gram dictionary here revolves around the idea that any n-gram can be present in more than one URL e.g., the trigram 'oog' occurs in google.com and boogle.com and maybe a point of error in any or either of them. Thus we consolidate all errors at the n-gram level from different domains.

Given a URL, we first create its squat space. Second, for each URL in the squat space we create n-grams e.g., consider the URL *google.com* and a URL from its squat space *gooogle.com*. Using the Needleman-Wunsch algorithm [47], we align both domains giving us *goo-gle* and *gooogle*. If we take n=5 and create n-grams, we have *[goo-g,oo-gl,o-gle]* and *[gooog,ooogl,oogle]* respectively for the two URLs. From these, we select the n-grams that are different between the two URLs giving us pruned n-grams. We then store all such n-grams as key-value pairs. For keys, we append each corresponding n-gram pair of the base URL with that of the mutated URL from our pruned set. In our example, we append the first pair and create *[goo-g/gooog]*. We do this for all available pairs and each pair serves as a unique key. For each key the value is initialized at zero. Then, for each available key, we comb through our typo datasets to find its occurrences. For every real-world typo of 'google', if the typo contains our subject n-gram (from the key), we increment its value, and if not, we decrement its value. As an example, we may end up with 5 occurrences of the pair *[goo-g/gooog]* and 7 occurrences of *[o-gle/oogle]*. Doing this for all n-grams from all domains, we generate our dictionary. Whether the error occurs in a google.com typo or boogle.com typo, for a particular n-gram we update the corresponding n-gram pair.

2) Training Phase: We use Random Forest for our prediction model. A similar vector as that of the HQ model is created for each URL and all its typos from the squat space. The 32 features are discussed in Table II. As an example, if we have *google* and *boogle*, first we will calculate their features and have two vectors of length 32. We then subtract the two vectors to get a single *Difference Vector* (DV) of size 32. This vector will signify the features which changed in order to make the typo version possible.

Second, we compute all n-gram pairs between the base and the mutated URL. Third, we search through our dictionary for all pairs and append the values to the end of the DV. This gives us our input vector of size of 45 (if less then 45 we append zeros at the end). We make input vectors of all typos for all domains. Finally, we search through our dataset to see if the mutated URL actually occurs as a typo of the original URL. If it does the overall label of the corresponding DV for that URL is set to 1 otherwise 0. For instance, if our typo was *gooogle.com*, we would create a DV by subtracting the 32 length feature vector of *google* and *gooogle*. We will then append the values of the n-gram pairs from our dictionary. Finally, we will see if *gooogle* actually occurs as a typo of *google* and assign it the corresponding label, 1 or 0. All such vectors along with their labels are fed to the Random Forest classifier for training.

3) Testing Phase: For *facebook.com*, if we have a possible typo *facbook.com*, and we want to see its probability of occurrence in real world we'll test it through our trained Random Forrest classifier. Like in the training phase, we will create its DV against *facebook.com*. We will also extract its n-grams from our dictionary, and append their values at the end of the DV which will give us the final test vectors. We will pass this test vector to our Random Forest Classifier and it will output its likelihood of occurrence in real world, which is a number between 0-1. We can then sort the top k typos from the entire squat space and defensively register those.

VI. Findings and Results

We now present some interesting results pertaining to the Hardness Quotient, the prediction service and typosquatting in the wild.

Correlation Between Error Probability and HQ: We claim that the Hardness Quotient for a particular URL determines the difficulty incurred in typing it out. To validate this claim, we put our model to the test using real-world data. Ideally, URLs that are being mistyped the most should have larger values of the Hardness Quotient (HQ). To this end, we measure the correlation between our Hardness Quotient and actual error frequency by applying a held-out sample from the data collected through our keystroke study and the typed URL dataset to our model. We use Spearman's Rank Correlation Coefficient (due to its robustness to outliers) to determine the extent of correlation of the total number of typos for each word with its corresponding HQ value. Figure 1 shows a scatter plot of the variables involved. It illustrates a gradual, increasing trend of the error frequency as the value of HQ grows. The Spearman's coefficient value here is 0.87, which is quite decent given an actual sample from Internet users.

Typosquatting Saturation: Beyond determining the difficulty of typing domain names, we can also use our model to compute a metric of *saturation* within the squat space: that is, how many of the most lucrative typos have already been registered for a given domain name, and by whom. This investigation can let us know how well typo registrants (both offensive typosquatters and defensive services like MarkMonitor) are doing, both with respect to targeting the

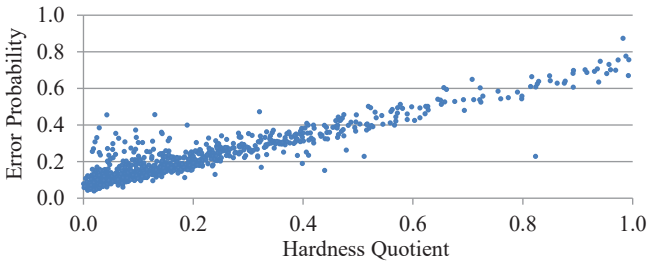

Fig. 1. Scatter plot showing correlation between probability of errors and the Hardness Quotient. The values are densely scattered along the diagonal meaning that as the HQ increases the resulting probabilities of error also increase proportionally.

Domain Echelon	Portion Registered
rank 1 to 10	98.8%
rank 11 to 100	88.0%
rank 101 to 1,000	48.7%
rank 1,001 to 5,000	28.6%

TABLE III
PORTION OF 40 MOST LIKELY TYPOS REGISTERED FOR .COM SITES AS A FUNCTION OF THEIR ALEXA RANK.

correct domains. To conduct this investigation, we enumerate the 40 most likely typos for each website within the 5,000 most popular domains in the .com TLD on Alexa. We cross reference this list with the `.com` domain name zone file provided by Verisign, which enumerates every registered .com domain name, along with the hostname and IP address of its authoritative nameservers.

Table III outlines the basic findings. Here we see that for the most popular websites, the typosquatting space is completely saturated, with only a few exceptions. Likewise, most of the 40 most likely typos are registered for the top 100 .com domains as well. The ratio drops and continues to do so for the remainder of the top 5,000 domains. Even the lowest ranked websites among this list have several hundred thousand regular users according to Alexa's data, indicating that there may yet be some value to be captured here.

Accuracy of Prediction: To test the accuracy of our prediction service, we measured how well we predict the more likely mutations of a base URL. A sorted list of our model's output was compared against the frequency of occurrence of each URL instance from the URL Fixer data to determine how much coverage of the typosquatted traffic the prediction service provides. The results of this experiment are shown in Figure 2. URLs such as Facebook and Google, which are characterized by enormous volumes of network traffic, are more prone to typosquatting as typosquatters can benefit more. As a result, the number of typosquatted versions of these URLs is relatively higher, which in turn implies that achieving near 100% coverage would require a large number of defensive registrations. However, our prediction service still provides decent coverage to these high-volume sites i.e., more than 60% while keeping the number of defensive registrations below a modest threshold of 10. On the other hand, URLs such as Amazon which have comparatively less traffic show

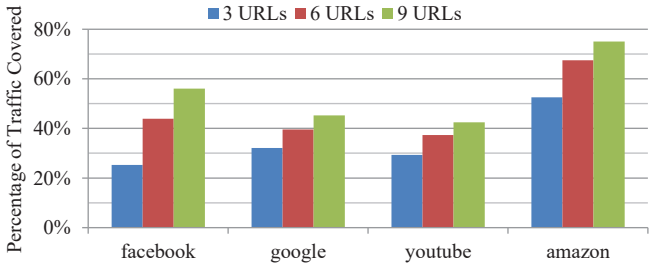

Fig. 2. Percentage of typosquatted traffic covered through predicted URLs

Method	Accuracy	Precision	Recall	F1 Score
Vanilla Features	86.46	82.41	74.37	78.19
Features + 4-grams	95.7	94.3	92.4	93.38

TABLE IV
TABLE DEMONSTRATING THE INCREASE IN ACCURACY RESULTING FROM THE N-GRAM COMPONENT ON TEST DATA

close to 80% coverage through our prediction service. The case of Youtube warrants added explanation here. As there are numerous video streaming websites, which have the substring 'tube' as part of their legitimate domain names particularly those in the adult domain, it is difficult to predict mistyped versions of Youtube that typosquatters can abuse as the squat space is already saturated. These legitimate websites and their corresponding typos mean that the accuracy of our prediction service has to suffer i.e. close to 40%. Barring a few special cases like Youtube, the overall results here show that our prediction service makes accurate predictions and provides a high degree of coverage to URL owners, which substantially reduces the likelihood of being abused by typosquatters. The average percentage of coverage for the top 20 Alexa websites for 3 defensively registered URLs is 35%, for 6 URLs is 46% and for 9 URLs is 60%. The overall results are demonstrated in Table IV.

VII. DISCUSSION AND LIMITATIONS

We now highlight some limitations of our work and discuss a few additional aspects.

Spelling Mistakes vs Typos: One limitation of our work pertains to the issue of spelling mistakes. For a user who has a weak grasp of the English language, this may very well be the case. However, in our experience, most of the URLs are not direct words from the English language i.e. Google, Instagram, Wikipedia etc. and so the argument is slightly weakened, though still valid as often the substrings are borrowed from the language. Additionally, from the data we collected, it appears that the errors are not entirely random and exhibit patterns and structural regularities, such as keys being replaced by their neighbors only or being triple typed instead of double typed etc., which means that the mistakes are not spelling mistakes but have some relationship with the underlying typing medium. Finally, spelling mistakes become even less of an issue as over time, when the user has typed the same URL a few different times, his spelling mistakes tend to improve and the errors that he continues to exhibit are by and

large typographical ones.

Typing on Handheld Devices and Modern Keyboards: Handheld devices are becoming increasingly pervasive and so we now observe users typing on touch pads, touch screens and fancier touch-based keyboards. This implies that users will make typing errors resulting from completely different anatomical limitations substantially altering the underlying semantics of the typing model. Hence, there is need for further extension of this study for different typing media, such as those mentioned above and we plan on pursuing this as future work.

Issue of False Positives: One final limitation that we discuss is that of false positives. Some URLs that we classify as typos might actually be legitimate websites that are coincidentally closely "related" to another base URL however, this is a small fraction compared to the actual typo instances and our analysis should hold for a majority of the cases.

VIII. Conclusion

In this paper, we took a first of its kind, human-centered view of the typosquatting phenomenon. We conducted a fixed-text keystroke study designed to explore the type, frequency and patterns of URL-based typing errors. Using the data, we developed a Hardness Quotient for each word using a quantitative typographical model that attempts to capture the relationship between human anatomy, keyboard layouts and typing mistakes. We cross-validated our model and the hardness measure using actual data from mistyped URLs in the Internet. The results verify that certain semantic features present in a domain name make it more susceptible to errors and hence typosquatting. Finally, using the model we build a prediction service that outputs the top k URLs that business owners should also purchase along with the base URL to minimize the effects of typosquatting and incorporate the prediction service into a Firefox browser plugin.

References

[1] T. Moore and B. Edelman, "Measuring the perpetrators and funders of typosquatting," in *FC*, 2010.

[2] "There's no "i" in twtter: How to outsmart typosquatting," https://tinyurl.com/jjjtmpe.

[3] "Typosquatting sites 'wikapedia' and 'twtter' have been fined $300,000 by UK watchdog," https://tinyurl.com/z8n9t84.

[4] J. Spaulding *et al.*, "The landscape of domain name typosquatting: Techniques and countermeasures," *CoRR*, vol. abs/1603.02767, 2016.

[5] P. Agten *et al.*, "Seven months' worth of mistakes: A longitudinal study of typosquatting abuse," in *NDSS*, 2015.

[6] M. T. Khan *et al.*, "Every second counts: Quantifying the negative externalities of cybercrime via typosquatting," in *IEEE S & P*, 2015.

[7] "URL Fixer," http://urlfixer.org/.

[8] "Public Security Log Sharing Site," http://log-sharing.dreamhosters.com.

[9] M. Meiss *et al.*, "Ranking web sites with real user traffic," in *WSDM*, 2008.

[10] M. E. Whitman *et al.*, "Cybersquatting: A case of first come/first served or piracy on the cyber-seas?" *Information Systems Security*, vol. 8, no. 1, 1999.

[11] S. Wright, "Cybersquatting at the intersection of internet domain names and trademark law," *IEEE Communications Surveys and Tutorials*, vol. 14, no. 1, 2012.

[12] A. Banerjee *et al.*, "SUT: quantifying and mitigating URL typosquatting," *Computer Networks*, vol. 55, no. 13, 2011.

[13] "Cybersquatting: typosquatting - facebook's 2.8 million dollars in damages and domain names," https://tinyurl.com/hp8vvxn.

[14] J. Szurdi *et al.*, "The long "taile" of typosquatting domain names," in *USENIX Security Symposium*, 2014.

[15] F. Damerau, "A technique for computer detection and correction of spelling errors," *Commun. ACM*, vol. 7, no. 3, 1964.

[16] A. Levenstein, "Binary Codes Capable of Correcting Deletions, Insertions and Reversals," in *Soviet Physics-Doklandy*, vol. 10, 1966.

[17] "Is Exercise Really Medicine? An Evolutionary Perspective ," https://tinyurl.com/zovsfof.

[18] "Born to Rest - Harvard Magazine," https://tinyurl.com/jdoya26.

[19] "CarpalX Keyboard Layout Optimizer," http://mkweb.bcgsc.ca/carpalx.

[20] "Workman Layout: The Layout Designed with Hands in Mind," https://tinyurl.com/px6pz9d.

[21] "Keyboard Layout Analyzer," https://tinyurl.com/lxzosqt.

[22] "What do people type in the address bar?" https://tinyurl.com/3maaq3r.

[23] J. W. Ratcliff and D. E. Metzener, "Pattern matching: The gestalt approach," vol. 13, no. 7, 1988.

[24] A. Moore, R. Wells, and D. Ranney, "Quantifying exposure in occupational manual tasks with cumulative trauma disorder potential," *Ergonomics*, vol. 34, no. 12, 1991.

[25] N. M. Hadler, *Clinical concepts in regional musculoskeletal illness*, 1987.

[26] L. Hymovich and M. Lindholm, "Hand, wrist, and forearm injuries: The result of repetitive motions." *JOEM*, vol. 8, no. 11, 1966.

[27] J. E. Nelson *et al.*, "Finger motion, wrist motion and tendon travel as a function of keyboard angles," *Clinical Biomechanics*, vol. 15, no. 7, 2000.

[28] R. M. Szabo and L. K. Chidgey, "Stress carpal tunnel pressures in patients with carpal tunnel syndrome and normal patients," *The Journal of hand surgery*, vol. 14, no. 4, 1989.

[29] A. Hedge and J. R. POWERS, "Wrist postures while keyboarding: effects of a negative slope keyboard system and full motion forearm supports," *Ergonomics*, vol. 38, no. 3, 1995.

[30] H. Seradge *et al.*, "In vivo measurement of carpal tunnel pressure in the functioning hand," *The Journal of hand surgery*, vol. 20, no. 5, 1995.

[31] M. Fagarasanu and S. Kumar, "Carpal tunnel syndrome due to keyboarding and mouse tasks: a review," *International Journal of Industrial Ergonomics*, vol. 31, no. 2, 2003.

[32] P. J. Keir *et al.*, "Effects of finger posture on carpal tunnel pressure during wrist motion," *The Journal of hand surgery*, vol. 23, no. 6, 1998.

[33] W. Hünting and T. L. E. GRANDJEAN, "Postural and visual loads at vdt workplaces i. constrained postures," *Ergonomics*, vol. 24, no. 12, 1981.

[34] J. Fish and J. Soechting, "Synergistic finger movements in a skilled motor task," *Experimental Brain Research*, vol. 91, no. 2, 1992.

[35] D. R. Gentner, "Expertise in typewriting," *The nature of expertise*, 1988.

[36] Zatsiorsky *et al.*, "Enslaving effects in multi-finger force production." *Experimental brain research*, vol. 131, no. 2, 2000.

[37] M. H. Schieber and M. Santello, "Hand function: peripheral and central constraints on performance," *Journal of Applied Physiology*, vol. 96, no. 6, 2004.

[38] K. T. Reilly and G. R. Hammond, "Independence of force production by digits of the human hand," *Neuroscience letters*, vol. 290, no. 1, 2000.

[39] L. H. Shaffer, "Intention and performance." *Psychological Review*, vol. 83, no. 5, 1976.

[40] D. R. Gentner, "Evidence against a central control model of timing in typing." 1982.

[41] K. S. Lashley, "The problem of serial order in behavior," 1951.

[42] D. E. Rumelhart and D. A. Norman, "Simulating a skilled typist: A study of skilled cognitive-motor performance," *Cognitive science*, vol. 6, no. 1, 1982.

[43] D. A. Norman and D. Fisher, "Why alphabetic keyboards are not easy to use: Keyboard layout doesn't much matter," *Human Factors*, vol. 24, no. 5, 1982.

[44] A. Dvorak, "There is a better typewriter keyboard," *National Business Education Quarterly*, vol. 12, no. 2, 1943.

[45] K. Provins and D. Glencross, "Handwriting, typewriting and handedness," *The Quarterly journal of experimental psychology*, vol. 20, no. 3, 1968.

[46] "Grading Scheme," https://tinyurl.com/y9rpwj96.

[47] Saul B. Needleman and Christian D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *J. Mol. Biol.*, vol. 48, no. 3, 1970.