Intruder Detector: A Continuous Authentication Tool to Model User Behavior

Leslie C. Milton
Information Technology Laboratory
U.S. Army ERDC, Vicksburg, MS, USA
leslie.c.milton@usace.army.mil

Atif Memon
Department of Computer Science
University of Maryland, College Park, MD, USA
atif@cs.umd.edu

***Abstract*—This paper presents techniques to continuously authenticate users as they interact with web-based software. Unique behavioral footprints, indicating patterns of use for groups of users, are captured from web server log files and integrated into an n-gram model. These statistical language models provide sequences and sub-sequences of user interaction, ordering, and temporal relationships. When users interact with web-based software, their stored usage profile is compared to their current interactions. Deviations may indicate malicious activity. We use our innovative tool, *Intruder Detector* (ID) to generate the profiles. Afterwards, we use various measures-of-effectiveness techniques to understand the feasibility of our approach. Our empirical study shows that session length and the prevalence of user data significantly affect the model's ability to correctly classify users.**

***Index Terms*—behavioral modeling; continuous authentication; software security; n-grams, statistical language models**

I. Introduction

Many web-based applications rely on authentication methods that are reliable, convenient and secure. Username and password have been universally accepted by most applications to be the only form of authentication. Some systems require the use of long passwords that need to be changed frequently. They can be difficult to remember, create, and manage [8]. In addition to long passwords, passwords that are too short or lack complexity also pose a significant risk. In a study of over 3.3 million leaked passwords from North America and Western Europe, SplashData records "123456" and "password" as the top two passwords chosen by users [24].

Conventional methods of authentication lack the ability to continuously monitor the user and verify their identity, leaving the computer system vulnerable to malicious or unintended use while the user is logged-in [18]. To improve the authentication process for web-based applications, there must be a method to continuously verify the identity of a user. *Continuous User Authentication* (CUA) has been proven to solve this limitation. CUA techniques monitor, verify, and authenticate users during their entire session. CUA generates usage profiles and compares them to the user's stored profile. If user activity deviates from its normal pattern of usage, the system generates an alarm. CUA systems have user profiles that are customized for every application. This makes it difficult for attackers to know which actions will be detected as intrusive [20].

Many studies have used biometrics to continuously authenticate users by the use of cognitive fingerprints, eye scans, color of user's clothing, and face tracking [18] [19] [5]. However, additional resources, such as hardware and cost, are needed to operate efficiently. Behavioral modeling addresses these limitations by monitoring how users interact with the system. Evaluating mouse movement, how users search for and select information, and the habitual typing rhythm of users are measures used to continuously observe a user's behavior [23][17]. Although these approaches do not require special hardware, most of them require the installation of specialized monitoring software.

This paper addresses challenges that occur when modeling the behavior of users that interact with web-based organizational information system applications. These applications run inside a Web browser-based front-end and are accessible via hypertext transfer protocol (HTTP). It also includes middleware to implement business logic and a back-end database. We categorize information system as organized systems for the collection, organization, communication and storage of information [25]. We use our innovative tool, *Intruder Detector* (**ID**) [16], to model unique behavioral footprints for each user. Patterns of use for a specific user or group of users is captured in this footprint and leveraged to "continuously" authenticate the user. **ID** performs behavioral analysis for each user and builds a context of each user's behavior, based on a statistical language model, to verify the user's identity. No additional hardware is required to deploy this tool. We conduct performance assessments to evaluate the feasibility of our approach using the the following metrics; recall, precision, F-measure, false positive rate, sensitivity, and specificity.

We provide the following contributions from this research:

- We develop a novel keyword abstraction technique to pre-process large volumes of web logs by eliminating incomplete, noisy and inconsistent data.
- We use n-gram language models to capture the behavior of users while they interact with organizational web-based applications.
- We develop a CUA framework with the ability to categorize user sessions into a predefined set of usage profiles.
- We introduce a preliminary set of evaluation metrics to test the feasibility of our approach.

The rest of the paper is organized as follows: Section II provides details of related work. Section III defines the basis

of CUA using natural language models and user categorization techniques. Section IV describes our empirical evaluation. Section V outlines significant findings and opportunities for future work. The conclusion is discussed in Section VI.

II. Related Work

Cybersecurity has become a key concern for many organizations and companies. For example, the Office of Personnel Management (OPM) informed millions of government and military employees that their personal information may be compromised [11]. For many systems, the first line of defense is authentication. Google and DARPA agree that elaborate password rules must be abandon and the use of strong authentication should be used to avoid impersonations [7] [6].

Over twenty years ago, when e-commerce and secure web was first introduced, passwords were mainly a stopgap measure. It was expected that something better would replace it soon. As applications and devices evolve, this means of authentication is becoming insufficient. CUA fills this gap by transparently monitoring user activity in an effort to identify deviations from normal workflow patterns. These patterns are stored usage profiles of each user of the system.

A robust CUA system has the following basic characteristics [10]:

- **Continual**: Re-authentication should be performed periodically to check if the current user is the logged-in user.
- **Non-intrusive**: Intrusive authentication hinders usability and provides a negative experience for the user. Therefore, the system must provide a seamless, non-intrusive user-friendly environment.
- **Behavioral**: The system must extract behavioral attributes from normal user operations. These attributes should be cost-effective and have unique usage profiles for each user.

The realm of CUA has been extensively evaluated with the use of biometrics. One study uses cognitive fingerprints to measure computational behavior by means of computational linguistics and structural semantic analysis [6]. This study uses a combination of metrics that include eye scans and keystrokes to evaluate how the user searches for and selects information. In addition, a number of CUA research studies use one or more hard and soft biometric traits to continuously authenticate a user. Niinuma *et al.* propose a CUA framework to automatically register the color of a user's clothing and their face as soft biometric traits [18][19]. Results from this study show that the system is able to successfully authenticate the user with high tolerance to the user's posture. Limitations to these studies exist because of the additional hardware that is needed to implement this technique which can become costly if an entire organization uses this feature to authenticate users.

Altinok *et al.* propose a continuous biometric authentication system that provides an estimate of authentication certainty at any given time, even in the absence of any biometric data [1]. In this case, the authentication uncertainty increases over time which leads to a decrease in system usability. In a similar study, Kang *et al.* introduce temporal integration of biometrics and behavioral features to continuously authenticate users [12].

In recent years, mobile devices have been used to learn user behavior. Researchers introduced SenSec as a mobile framework to collect sensory data to construct a gesture model of how a user interacts with a mobile device [28]. Similar to our work, n-grams are used to capture user patterns. The SenSec system achieves over 70% accuracy in user classification and authentication tasks. In addition, Saevanee *et al.* use multi-model biometric techniques with mobile devices using linguistic profiling, keystroke dynamics and behavioral profiling for user authentication [21]. Results from this study show a 91% reduction rate in the number of intrusive authentication requests.

This body-of-work extends beyond the aforementioned research studies in the following ways:

1) Instead of using traditional biometric traits, we explore the possibility of using log information that is naturally generated by web applications to improve usability through non-intrusive, transparent authentication.
2) This approach, integrated into a tool, uses a novel and simple n-gram language model to capture user behavior.
3) Experiments are based on data from actual users of a fielded Department of Defense (DoD) system who are completing day-to-day tasks.

Natural Language	**Web Behavior Language**	**Example**
Word	Link selection	View profile
Phrase	View	Watch online training video
Sentence	Action	Search archived files
Paragraph	Activity	Course registration
Document	Event	Prepare course evaluation report

TABLE I
Comparison of Natural Language and Web Behavior

III. Continous User Authentication

Web-based applications have rich sets of information that can be used to model user behavior, such as web server logs, database logs, graphical user interface accesses, etc. In our study, we focus on information captured in web server log files to predict user behavior. Web server logs capture all requests made to the server. For many systems, web server log files are not fully utilized. These logs, also called access logs, include historical information about the activities performed by users.

A. Web Behavior as a Language

It is important to understand individual and role-based behavior to detect common patterns. Specifically, we would like to use web logs to build usage profiles for each user. Grammars can be defined for human behavior and natural language [27] [3]. Table I illustrates the commonalities between natural language and web-based user behavior. Links or buttons in a web application represent a basic level of vocabulary for web behavior. A series of link clicks give

the user the ability to *view* various portions of the website (e.g., course videos). Meaningful sequential link selections may lead to various *actions*, *activities*, or *events*. Since web logs represent sequential actions, they can be encoded as sequences of symbols and used with a standard NLP technique to build computational usage models [14]. In this research, we experiment with n-grams to model sequences of keywords, each of which represent a user action. We use n-grams, derived from Markov models, to understand similarity in user actions to classify the user's identity. These generative models can learn each category of users then classify the users based on the generated knowledge. This method gives us the ability to perform user authentication task with only positive training samples.

Web server logs are preprocessed to removed unwanted entries. This helps validate the data captured in a users session [4]. This process cleans the data, identifies users and sessions, and generates keywords. The keywords are user behavior labels for the n-gram modeling process.

B. n-gram Models

n-gram models are Markov models which use $(N-1)$ elements of context to define the current state of the model [22]. These stochastic process models are mostly stationary since we are assuming past behavior is a good prediction of what will happen in the future. However, natural language is not stationary because the probability of upcoming words can be dependent on events that are arbitrarily distant and time dependent. Therefore, the statistical models of n-grams only give an approximation of the correct distributions and entropies of natural language. Constructing or training an n-gram model requires the ability to observe example sequences occurring in the domain to be modeled. To train a model well, single events from sequences in all relevant contexts must be observed. We compute probabilities based on a set of given observations. The observations are mapped to a series of class labels $\{w_0, w_1, w_2, ..., w_n\}$. Applying the chain rule of probability theory yields the probability of a sequence according to some prior context available at each data point:

$$\begin{aligned} P(w_1^n) &= P(w_1)P(w_2|w_1)...P(w_n|w_1w_2...w_{n-1}) \\ &= P(w_1)P(w_2|w_1)P(w_3|w_1^2)...P(w_n|w_1^{n-1}) \\ &= \prod_{k=1}^{n} P(w_k|w_1^{k-1}) \end{aligned} \tag{1}$$

C. User Categorization

A percentage of keywords are reserved based on a data split to represent the testing set, E, and use the remaining data as the training set, R, to train an N order n-gram model for the specified category. During the test phase of our experiments, we assign a probability to a sequence of events. We use binary categorization to judge a sequence as having likely been generated by a specific model (PASS) or not (FAIL). We introduce a probability threshold, t, for this pass/fail type of judgment for a sequence. Any sequence whose probability exceeds this threshold should be considered as a PASS, $+1$, and otherwise considered FAIL, -1.

A decision rule is used to predict the class membership of a given sequence of behavioral keywords, K. When new samples are encountered, the following decision rule is used:

$$\begin{cases} P(K,m) > t, & \text{then } y = +1 \\ P(K,m) < t, & \text{then } y = -1 \end{cases} \tag{2}$$

where $P(K,m)$ is the probability the behavioral keyword sequence is generated by the mth user's n-gram model. The probabilities are estimated using a training set of labeled data, $\{(m_0,y_0),(m_1,y_1),(m_2,y_2),...,(m_n,y_n)\}$, where label $y_i = \pm 1$ and depends on the class of m_i.

A more complex scheme that can also be useful for continuous authentication is multi-class categorization [15] [9]. For effective evaluation, this categorization method requires more training and test data. Under this decision-making approach, we can score an input sequence according to one of many models, and categorize the sequence as belonging to the model which estimates the highest probability. Therefore,

$$u = \arg\max_m P(K,m) \tag{3}$$

We use binary categorization by a simple threshold and multi-class categorization by comparing probabilities to translate n-gram models' estimations of sequence probability into decisions. We investigate our ability to make accurate decisions of various types as supported by these algorithms.

IV. Empirical Evaluation

In our previous work [16], we investigated a DoD online training website where each user account has an associated role (*Management (Mgmt), Technologist (Tech), Administrator (Admin), and User*). The dataset was generated based on the training and test data splits identified in Table II.

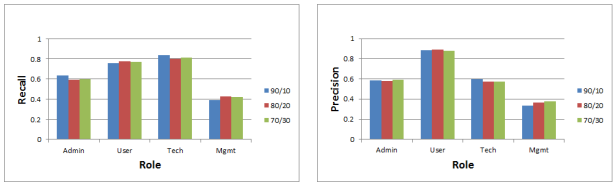
From this data, **ID** was able to build discriminating user roles, recognize various legitimate users operating in the same user session, and identify outliers in user behavior. This study extends beyond the aforementioned study by investigating the utility of **ID** when evaluating various metrics; *precision, recall, F-measure, false positive rate*. These metrics show the extent to which **ID** can be used with real-world systems.

1) Multi-class Categorization: We evaluate **ID**ś performance using $N = 2,3,4,5$ for each metric and class label (i.e., roles). We only show bi-gram (N=2) results since **ID** performs best when using this model.

Recall measures the proportion of positive examples that are correctly identified as positive. Precision measures the fraction of predicted examples that are relevant. After categorizing the roles using **ID** the *Mgmt* role yields the least recall and precision rates (approx. 40%) due to its limited prevalence in the dataset. The Tech role has the highest prevalence and yields the best recall rate but is not as precise when compared to the User role. The User role has the highest precision rate at approximately 90% with less than 60% precision for the Tech role. Therefore, the classifier is more precise when

Data Split	Role	Sessions	Keywords
90/10	Management-train	247	5045
	Management-test	28	814
	Technologist-train	421	31965
	Technologist-test	47	3709
	Admin-train	521	22645
	Admin-test	58	3388
	User-train	1735	24849
	User-test	193	2808
80/20	Management-train	220	4643
	Management-test	55	1216
	Technologist-train	374	28629
	Technologist-test	94	7045
	Admin-train	463	20298
	Admin-test	116	5735
	User-train	1542	21949
	User-test	386	5708
70/30	Management-train	192	3787
	Management-test	83	2072
	Technologist-train	327	24643
	Technologist-test	141	11031
	Admin-train	405	18122
	Admin-test	174	7911
	User-train	1349	19318
	User-test	579	8339

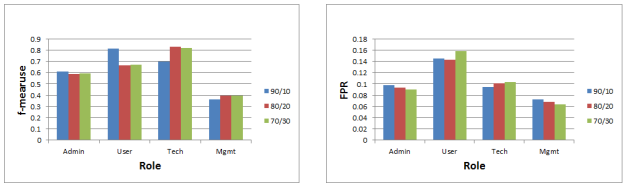
TABLE II
NUMBER OF KEYWORDS AND SESSIONS FOR EACH DATA SPLIT



(a) Recall (b) Precision

Fig. 1. Comparison of Recall and Precision

predicting roles that have more per-session data available (i.e., User role). Historically, precision and recall have been used as classification performance metrics. In many machine learning research papers, precision is more informative for non-binary classifiers. During our analysis, we determined which metric or combination of metrics is most informative for the task of identifying user roles. In some cases, it is very difficult to assess the performance of precision and recall separately. We use the F-measure as the weighted harmonic mean of precision and recall.

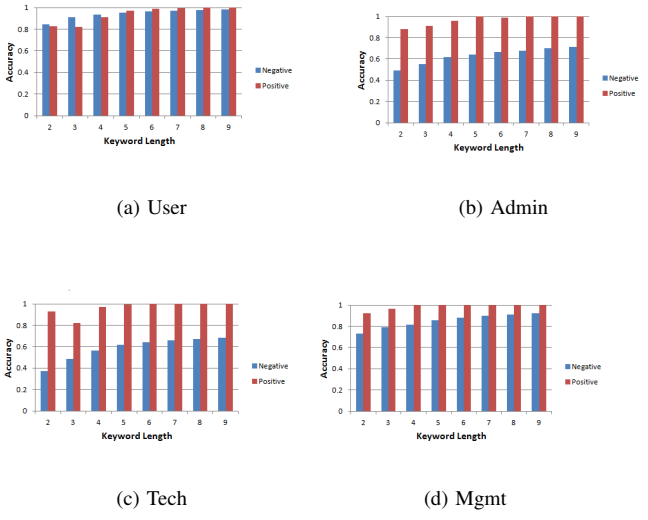


(a) F-measure (b) False Positive Rate (FPR)

Fig. 2. F-measure and False Positive Rate

The F-measure for the User role is 81% with 14% false positive rate (90/10 data split only). Since this category of users has the largest number of training sessions, bi-grams with the 90/10 data split may have better performance when more per-session behavior from each user is present in the dataset. The F-measure for the Tech role is approximately 83% for bi-grams with the 80/20 and 70/30 data splits (with 10% FPR). The Tech role has the largest number of keywords used for training with 59% prevalence. Generally, the larger the training set, the better the classifier for this category of users. As expected, the Mgmt role has the lowest F-measure for each model. All data splits for this category perform at or below 40%. To train a category of users well, there is a noticeable difference in the number of keywords needed. For example, higher F-measure rates were received for users with approximately 850 keywords and 120 sessions (90/10 data split) and roles with approximately 30,000 keywords and 400 sessions (80/20 and 70/30 data split). The User role has the highest FPR for bi-grams using each data split with 70/30 data split having a slightly higher FPR (approximately 16%). It is important to emphasize that this data set has only 14% of keyword data labeled as User and 59% labeled as Tech. Since the Tech role is more prevalent than all roles combined, we observed an increased FPR for this role.

2) Binary Categorization: Binary categorization is used to detect uncharacteristic activity based on outliers in user behavior. Because this task offers only two possible classes, a random model with no observation would achieve 50% accuracy in categorizing user activity as PASS or FAIL according to each role-specific model. We expect **ID** to outperform this random baseline.



(a) User (b) Admin

(c) Tech (d) Mgmt

Fig. 3. Role-based Binary Classification

This task uses models independently. Specificity (i.e., *1-FPR*) and sensitivity (i.e., recall) are statistical measures

used for the performance of this classification task. Both training and test data from the remaining three roles are used to evaluate the model's ability to reject uncharacteristic sequences (i.e., negative examples) by measuring specificity. Only test data from the model's own role is used to evaluate its ability to accept valid sequences (i.e., positive examples) using sensitivity measures. Additionally, we consider the affect of test sequence length on the performance of this task by evaluating the number of keywords of an input sequence. We adapt the model's output probability to be an indicator of entropy, a measure of uncertainty in a system. This provides the ability to normalize by the length of the input sequence. By doing so, a single threshold value for the binary classification task is maintained. Binary classification results are show in Figure 3.

Efficient experimental results are observed when accepting or rejecting snippets of user sessions based on role. Each model was tested against every available subsequence of user keywords, from lengths two to nine with a probability threshold of -0.6. Best performance was observed when using N=9. Note that accuracy is plotted separately for positive and negative test samples to analyze the models ability to reject uncharacteristic sequences (i.e., specificity) and accept valid sequences (i.e., sensitivity).

As expected, the length of sessions provided to models significantly affects the ability to correctly classify the example. The effect of length was much greater on negative examples, as failures due to rejecting a positive sample are rare after lengths greater than four. The User model has the most sessions of activity and performs at a level greater than 90% on all samples for lengths greater than three. When rejecting uncharacteristic sequences, the Management model eventually achieved performance of 92%, though this took sessions of length nine. With a 9% prevalence rate, it is evident that this role needs more training/test data for the identification of outliers.

V. Discussion and Future Work

It's important to identify the prevalence of labeled data by determining how often a particular label occurs in the dataset. The dataset under evaluation is heavily populated with Tech-level keywords (approximately 59%). Therefore, the weighted harmonic mean of precision and recall (i.e., F-measure) is highest for this particular role at approximately 83% for each data split. For the multi-class approach, bi-grams seem to have the most stability but its evident that more training data is needed to reduce the false positive rate.

From our results, one can conclude that binary classification proves to be much more effective than a random baseline at detecting uncharacteristic user behavior. Due to a large data set and elevated level of privileges for tasks, it was expected that the Tech user role, for binary categorization, would have one of the strongest models but this was not observed. With the multi-class categorization approach, we observed below optimal categorization rates for the User role model. Conversely, the User model, in the binary approach, is stronger. This improvement in performance could be due to the use of shorter sessions which are less likely to contain unseen events. In this case, we rely on the validity of smoothing assumptions for accurate probability estimation [13].

Throughout this study, it has been evident that many advantages and disadvantages exist for statistical language models. These machine learning models contain flexible training and include test data that can be used to update execution strategies. If appropriate training is available, our models are capable of detecting malicious or unintended usage. In contrast, if relevant training is not available, poor prediction performance is unavoidable. This approach is also highly dependent on assumptions made by a system administrator. Therefore, detection performance can be affected by slight changes in the keyword abstraction process.

There are various ways our work can be extended to explore this research domain. We conclude that with two of the four role-based models considered, the correct identification of negative samples was above 90% as well as a 100% correct acceptance of positive samples. These findings are promising, and motivate future work to better understand model-specific differences which make this task more difficult for some cases (e.g., Admin and Tech roles) than others. In particular, the finding that User sessions can so easily be protected against uncharacteristic usage is promising.

Many web applications have different sensitivity levels for security threats. Some systems must be protected at higher levels than other systems. There is a need to investigate an adaptive scheme (i.e., tunable parameter) to determine thresholds for different applications. The thresholds may be adjusted based on a user's role, application infrastructure, business logic, usage patterns, etc.

In many settings, statistical language models are computationally intensive. Building n-grams over large datasets pose challenges to memory and speed [2]. The computational cost of running these models should be calculated in real time to access the current usability performance. There may be a need to utilize a high performance computing environment to increase prediction time in an effort to help prevent or interrupt malicious web use.

There are various performance metrics available for classification tasks. The performance metrics used in this work can be extended to include the time it takes to identify legitimate and malicious use. When outliers are identified, it's important to detect this activity before the active session is complete. However, in some cases, this time metric may allow some flexibility to help application forensics analyst gather reliable evidence to use against perpetrators.

Finally, our work uses a single modality (i.e., web server logs) for CUA tasks. The use of a multi-modal approach may prove to provide a more transparent authentication process and help reduce false positive rates. Even though our models are able to detect legitimate users and outliers, the false alarms can be annoying and decrease usability dramatically.

VI. Conclusion

Many organizations are becoming increasingly aware of their security posture. It's important to identify cybersecurity-related risks and develop strategies to mitigate them. Continuous user authentication is one approach that can be utilized to identify impersonations and misappropriation of authentication credentials [26]. Specifically, this technique can be used with web applications to provide reliable and secure authentication.

In this work, we present an approach to address the need for web-based continuous user authentication. We employ the use of n-grams to capture user interaction with web-based software. After learning the behavior, we illustrate the ability to classify users and identify uncharacteristic behavior using multi-class and binary categorization. Results show model-specific differences in user behavior with performance highly dependent on session and keyword size. Our CUA implementation is continual, non-intrusive, and behavioral. We limit this study to web-based organizational information systems. However, the CUA domain can be extended to context-aware computing, Internet of Things (IoT), and people-centric applications.

Acknowledgment

The authors would like to thank the U.S. Army Engineer Research and Development Center (ERDC). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

References

[1] Alphan Altinok and Matthew Turk. Temporal integration for continuous multimodal biometrics. In *In Multimodal User Authentication*, pages 131–137, 2003.

[2] Lars Bungum and Bjorn Gamback. Efficient n-gram language modeling for billion word web corpora. In *Workshop on Challenges in the Management of Large Corpora*, LREC '12, 2012.

[3] Kenneth Burke. *Language as Symbolic Action*. University of California Press, 1966.

[4] Robert Cooley, Bamshad Mobasher, and Jaideep Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1:5–32, 1999.

[5] DARPA. Active authentication: http://www.darpa.mil/program/active-authentication, 2013. [Online; accessed 24-November-2015].

[6] I. Deutschmann, P. Nordstrom, and L. Nilsson. Continuous authentication using behavioral biometrics. *IT Professional*, 15(4):12–15, July 2013.

[7] Eric Grosse and Mayank Upadhyay. Authentication at scale. *IEEE Security and Privacy*, 11:15–22, 2013.

[8] Richard P. Guidorizzi. Security: Active authentication. *IT Professional*, 15(4):4–7, 2013.

[9] Paul Honeine, Zineb Noumir, and Cdric Richard. Multiclass classification machines with the complexity of a single binary classifier. *Signal Processing*, 93(5):1013 – 1026, 2013.

[10] Harini Jagadeesan and Michael S. Hsiao. Continuous authentication in computers. *Continuous Authentication using Biometrics: Data, Models, and Metrics*, 1:40–66, 2012.

[11] The Wall Street Journal. U.s. suspects hackers in china breached about 4 million people's records, officials say: http://www.wsj.com/articles/u-s-suspects-hackers-in-china-behind-government-data-breach-sources-say-1433451888, 2015. [Online; accessed 17-November-2015].

[12] Hang-Bong Kang and Myung-Ho Ju. Multi-modal feature integration for secure authentication. In *Proceedings of the 2006 International Conference on Intelligent Computing - Volume Part I*, ICIC'06, pages 1191–1200, Berlin, Heidelberg, 2006. Springer-Verlag.

[13] R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 181–184 vol.1, 1995.

[14] Jimmy Lin and W. John Wilbur. Modeling actions of pubmed users with n-gram language models. *Inf. Retr.*, 12(4):487–503, August 2009.

[15] Yiguang Liu, Zhisheng You, and Liping Cao. A novel and quick svm-based multi-class classifier. *Pattern Recogn.*, 39(11):2258–2264, November 2006.

[16] Leslie Milton, Bryan Robbins, and Atif Memon. N-gram based user behavioral model for continuous user authentication. In *The Proceedings of the Eighth International Conference on Emerging Security Information, Systems, and Technologies (SECURWARE 2014)*, 2014.

[17] Fabian Monrose and Aviel D. Rubin. Keystroke dynamics as a biometric for authentication. *Future Gener. Comput. Syst.*, 16(4):351–359, February 2000.

[18] Koichiro Niinuma, Anil K. Jain, Jain B.V.K.V Kumar, S. Prabhakar, and A. A. Ross. Continuous user authentication using temporal information. *SPIE.*, 7667, 2010.

[19] Koichiro Niinuma, Unsang Park, and Anil K. Jain. Soft biometric traits for continuous user authentication. *Trans. Info. For. Sec.*, 5(4):771–780, December 2010.

[20] Animesh Patcha and Jung-Min Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Comput. Netw.*, 51(12):3448–3470, August 2007.

[21] Hataichanok Saevanee, Nathan Clarke, Steven Furnell, and Valerio Biscione. Continuous user authentication using multi-modal biometrics. *Computers and Security*, 53:234 – 246, 2015.

[22] C. E. Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1):3–55, January 2001.

[23] Chao Shen, Zhongmin Cai, and Xiaohong Guan. Continuous authentication for mouse dynamics: A pattern-growth approach. In *Proceedings of the 2012 42Nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, DSN '12, pages 1–12, Washington, DC, USA, 2012. IEEE Computer Society.

[24] SplashData. Splashdata news, 2015. [Online; accessed 03-October-2015].

[25] Information System. Information system, 2015. [Online; accessed 03-October-2015].

[26] Issa. Traore and Ahmed A.E. Ahmed. Introduction to continuous authentication. In *IT Policy and Ethics: Concepts, Methodologies, Tools, and Applications*.

[27] James V. Wertsch. *Mind as Action*. Oxford University Press, 1998.

[28] Pang Wu, Joy Zhang, Jiang Zhu, and Xiao Wang. Sensec: Mobile security through passive sensing. In *Proceedings of the 2013 International Conference on Computing, Networking and Communications (ICNC)*, ICNC '13, pages 1128–1133, Washington, DC, USA, 2013. IEEE Computer Society.