Filtrado de SPAM en SMS mediante algoritmos de aprendizaje automático

Lenin J. Pin G.[1,2]

[1] Maestría en gestión estratégica de tecnologías de la información, Facultad de Ingeniería, Universidad de Cuenca, Av. 12 de Abril y Agustín Cueva, EC010112, Cuenca, Ecuador.
[2] Facultad de Ciencias Técnicas, Universidad Estatal del Sur de Manabí, Km. 1,5 Vía Noboa, Jipijapa, Manabí, Ecuador.

Autor para correspondencia: lenin.pin@ucuenca.ec, jonatan.pin@unesum.edu.ec

Fecha de recepción: 30 de julio de 2017 - Fecha de aceptación: 15 de agosto de 2017

RESUMEN

Una de las más comunes formas de comunicación a través de teléfonos móviles sigue siendo mediante SMS o servicio de mensajes cortos, por sus siglas en inglés. Las entidades financieras, televisoras y las propias operadoras de telefonía son ejemplos de compañías que aprovechan al máximo este tipo de comunicación, pero esta tecnología no está exenta de los molestosos mensajes no deseados o SPAM. El presente artículo describe la aplicación de algoritmos de aprendizaje automático como medio para la detección de SMS no deseados, y mediante la experimentación con un conjunto de datos de 5,574 mensajes de texto o SMS evalúa el rendimiento de modelos que utilizan técnicas como Regresión Logística, Super Vector Machine, KNN, RandomForest y AdaBoost para clasificar y predecir mensajes no deseados.

Palabras clave: Aprendizaje Automático, SMS, Super Vector Machine, Regresión Logística, KNN, RandomForest, AdaBoost.

ABSTRACT

One of the most common forms of communication using mobile phones is through SMS, or short message service. Financial institutions, television companies and the telephone operators are examples of companies that takes advantage of this type of communication; however, this technology is not exempt from unwanted messages or SPAM. This article describes both the application of automatic learning algorithms for SPAM's filter and an experimenting with a data set of 5,574 SMS to evaluate the performance of models using techniques such as Logistic Regression, Super Vector Machine, KNN, Random Forest and AdaBoost to filter and predict unwanted messages.

Keywords: Machine Learning, SMS, Super Vector Machine, Logistic Regression, KNN, RandomForest, AdaBoost.

1. INTRODUCCIÓN

La tecnología de comunicación móvil ha tenido un desarrollo acelerado y los mensajes de texto de tipo SMS se han convertido en una forma muy popular de comunicación, alcanzando un crecimiento exponencial en muchos países, lo cual a criterio de Zhang *et al*. (2016) ha traído consigo también una invasión de mensajes de tipo SPAM o no deseados. Los factores que propiciaron este crecimiento son: 1) la disponibilidad de planes de SMS a gran escala de bajo costo; 2) fiabilidad (se asegura que el mensaje llega al usuario del teléfono móvil); 3) baja probabilidad de recibir respuestas de algunos receptores desprevenidos; y 4) el mensaje puede ser personalizado. Según estudios de la GSMA, estos mensajes no deseados provocan molestias a los usuarios tales como pérdida de tiempo y consumo

innecesario de ancho de banda. Situación que también preocupa a los proveedores de servicios ya que perturba a sus clientes o incluso hace que pierdan suscriptores.

Las tecnologías de descubrimiento de spam en SMS provienen del filtrado de correo electrónico no deseado, pero existen características únicas para la detección de SMS no deseados como, por ejemplo, el filtrado en tiempo real mientras que en los correos electrónicos puede ser fuera de línea. Según los antecedentes encontrados en el estudio de Yu & Chen (2012), la detección spam en SMS puede agruparse en filtros basados en contenidos o en redes sociales. Los investigadores de métodos de filtrado o clasificación de mensajes de spam SMS se enfrentan al reto de poder acceder a un conjunto de datos que permita realizar con éxito sus investigaciones. Según Abdulhamid *et al*. (2017), esto también se da en la evaluación de nuevas propuesta de métodos de filtración y detección de spam SMS.

Las técnicas para detectar, filtrar o clasificar mensajes de texto no deseados, a criterio Abdulhamid *et al*. (2017), están diseñadas para funcionar en la capa de acceso (AL) o capa de usuario final y otras en la capa de proveedor de servicios (SPL).

Según lo documentado por Guzella & Caminhas (2009), la información contenida en un mensaje se divide en el encabezado (campos que contienen información general sobre el mensaje, como el sujeto, el remitente y el destinatario) y el cuerpo (el contenido real del mensaje). Antes de que la información disponible pueda ser utilizada por un clasificador en un filtro, se requieren pasos de preprocesamiento apropiados. La Figura 1 muestra los principales pasos para realizar una técnica de filtrado y se pueden presentar como:

1. Tokenización, que extrae las palabras en el cuerpo del mensaje;
2. Lematización, reduciendo las palabras a sus formas de raíz (por ejemplo, "‘extracting" a "extract");
3. Eliminación de palabra de parada, eliminando algunas palabras que a menudo ocurren en muchos mensajes (por ejemplo, "to", "a", "for");
4. Representación, convierte el conjunto de palabras presentes en el mensaje en un formato específico requerido por el algoritmo de aprendizaje de máquina utilizado

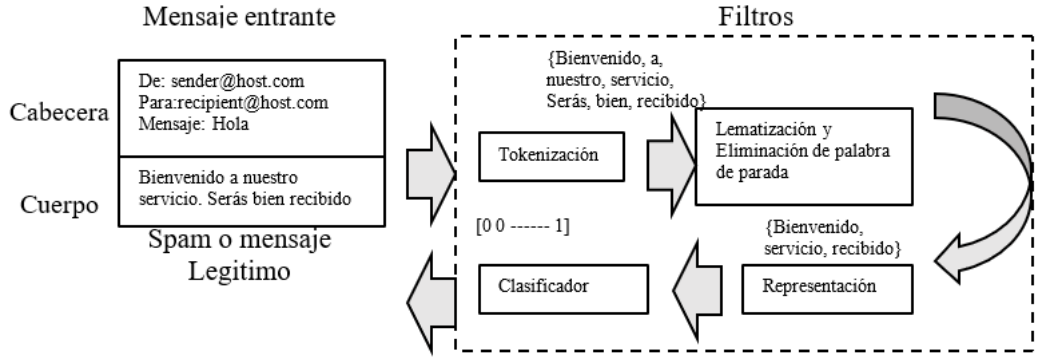

Figura 1. Pasos principales aplicados en las técnicas de filtrado de SPAM en SMS (Guzella & Caminhas, 2009).

En la Figura 1 se considera que sólo la información contenida en el cuerpo del mensaje es utilizada por el filtro, ya que el preprocesamiento de los encabezados requiere procedimientos específicos, dependiendo de los campos considerados. Además, se supone que los contenidos del mensaje han sido decodificados antes del análisis por el filtro, como es requerido para algunos mensajes con ciertas codificaciones de caracteres.

Según lo publicado por Almeida & Alberto (2013); Almeida, Gómez Hidalgo, & Yamakami (2011), para realizar el análisis de los mensajes se separan los mensajes en tokens, como se indica:

- token1: empiezan con cualquier carácter imprimible, seguido de cualquier carácter numérico o alfanumérico; pero no incluye puntos ni comas en la mitad.
- token2: cualquier secuencia de caracteres separadas por espacios en blanco, tabulaciones, puntos, comas y guiones.

Adicionalmente, 70% de los mensajes se separaron para entrenamiento (3,900) y el resto (30%) para pruebas (1,674). En la Tabla 1, se muestran los mejores resultados obtenidos en artículos previos.

Tabla 1. Resultados obtenidos en (Almeida & Alberto, 2013; Almeida *et al*., 2011; Almeida *et al*., 2013).

Artículo	Clasificador	SC% Spam Capturados	BH% Legítimos bloqueados	Acc% Exactitud	MCC Coeficiente de correlación de Matthews
[1]	SVM Lineal + token1	83.1	0.18	97.64	0.893
[1]	AdaBoost + token2	84.48	0.53	97.5	0.887
[1]	1NN + token2	43.81	0.00	92.7	0.636
[2]	Logistic Regression + token2	95.48	2.09	97.59	0.899
[3]	Random Forest + token2	65.23	0.12	95.36	0.782
[1]	SVM Lineal + token1	83.1	0.18	97.64	0.893

Leyenda:
[1] Almeida, Gómez Hildago, & Yamakami (2011). Contributions to the study of SMS spam filtering: New collection and results.
[2] Almeida & Alberto (2013). Learning to block undesired comments in the blogosphere.
[3] Almeida, Gómez Hidalgo, & Silva (2013). Towards SMS spam filtering: Results under a new dataset.

Existen algunos problemas principales para el desarrollo del algoritmo de filtrado de SMS, como el bajo número de características que se puede extraer de los mensajes, en comparación a los correos electrónicos, la corta longitud de los mensajes, y su lenguaje informal lleno de modismos y abreviaturas.

El objetivo del presente trabajo es aplicar diferentes algoritmos de aprendizaje automático al problema de clasificación de spam SMS, evaluar su rendimiento y compararlos con los resultados reportados en la literatura. Este artículo se encuentra organizado de la siguiente manera: luego de esta introducción, en donde se presenta una breve revisión de trabajos que han analizado el problema descrito, en la sección dos se describe el conjunto de datos utilizados y los algoritmos de aprendizaje automático que se han seleccionado para comparar su rendimiento, en la sección tres se detalla el resultado de la experimentación desarrollada, en la sección cuatro se analizan los resultados obtenidos, y, finalmente, en la sección cinco se presentan las conclusiones y las propuestas de futuros trabajos a desarrollar.

2. MATERIALES Y MÉTODOS

2.1. Descripción del conjunto de datos

El conjunto de datos “SMS Spam Collection” es un conjunto de 5,574 mensajes de texto SMS (Short Message Service) clasificados como Ham (legítimo) o Spam (no solicitado, no deseado). Fue compilada por Almeida, Gómez Hidalgo y Yamakani, tomando como fuente: 425 mensajes de la página web Grumbletext, 3,375 mensajes aleatorios escogidos de NUS SMS Corpus (departamento de ciencias de la computación de la Universidad Nacional de Singapur, 450 mensajes recogidos por Caroline Tag para su tesis doctoral y 1,324 mensajes de SMS Spam Corpus v.0.1 Big.

Para la ejecución de los algoritmos seleccionados, se tomaron el 70% de los mensajes para entrenamiento y el 30% restante para pruebas. Las cantidades respectivas de mensajes se muestran en la Tabla 2.

Tabla 2. Estadísticas básicas.

Mensaje	Cantidad total	%	Muestras de entrenamiento	Muestras de prueba
Ham	4,827	86.60	3,378	1,449
Spam	747	13.40	523	224
Total	5,574	100	3,901	1,673

2.2. Metodología

- En el conjunto de datos mencionado, se clasificaron en forma binaria las clases: Spam como 1 y Ham como 0. Posteriormente se utilizó la técnica TF-IDF (Term frequency – Inverse document frequency), para crear una matriz con estas características para los tokens reconocidos en cada mensaje. Sobre esta nueva matriz se aplicaron los algoritmos seleccionados.
- Para poder comparar los resultados de los experimentos con los artículos anteriores, se utilizó las mismas métricas de calidad indicadas por Almeida & Alberto (2013) y Almeida *et al* (2011, 2013).
  - Spam Caught(SC%): proporción de muestras tipo spam clasificadas correctamente, TP/(TP+FN).
  - Blocked Ham(BH%): proporción de muestras tipo ham clasificadas incorrectamente, FP/(TN+FP)
  - Exactitud (Acc%): (TP+TN)/(TP+FP+TN+FN)
  - Coeficiente de Correlación de Matthews(MCC):

$$MCC = \frac{\mathrm{TP\ x\ TN} - \mathrm{FP\ x\ FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}}$$

MCC se utiliza como una medida de la calidad de las clasificaciones binarias. Devuelve un valor real entre -1 y+1. Un coeficiente igual a +1 indica una predicción perfecta; 0, una predicción aleatoria media; y-1, una predicción inversa. Se considera como una medida correcta cuando los tamaños de las clases son muy diferentes, como en este caso que la proporción es de 13.4% de mensajes no solicitados (spam) y 86.4% de mensajes legítimos (ham) (Almeida & Alberto, 2013).

A continuación, se listan los algoritmos utilizados y la estrategia para su ajuste.

- Modelo de regresión logística utilizando las características de mensajes aleatorios.
- Support Vector Machine (SVM) al conjunto de datos, en ella se emplearán kernel Lineal, núcleo polinomial (se aumentarán los grados del polinomio de grado 2 al 4), función de base radial (RBF) y sigmoide. En los resultados del artículo de Shirani-Mehr (2013) se obtuvo un mejor tasa de error general para el Kernel Lineal con un 93.8% de spams capturados. En este trabajo se compara dichos resultados con los de la presente replicación experimental.
- K-nearest neighbours (kNN) para el conjunto de datos sobre la base del voto mayoritario de sus k vecinos más cercanos, donde se modificará su valor para predecir los resultados de la validación al conjunto de datos. Los valores que tomará el valor de k será 2, 10, 20, 50 y 100.
- Utilizaremos la implementación de Random Forest en la biblioteca de Scikit-Learn para promediar las predicciones probabilísticas y usaremos dos números de estimadores de 10 y de 100 para comparar los resultados de SC, BH y MCC. Este modelo ha dado buenos resultados según otras experimentaciones investigadas como en el caso explicado en Bin *et al*., (2016).
- AdaBoost construye secuencialmente clasificadores que se modifican a favor de instancias mal clasificadas por clasificadores anteriores. En cada iteración se aplicarán ciertos pesos a las muestras de entrenamiento, luego se incrementarán los pesos después de cada iteración para las clasificadas erróneamente según el modelo actual y los pesos. Al igual que el anterior algoritmo, utilizaremos dos números estimadores de 10 y 100 para comparar los resultados de SC, BH y MCC.

La estrategia de validación a empleada para el ajuste de hiperparámetros es la búsqueda por cuadrícula, *grid search* en inglés, junto a la validación cruzada, las cuales son los métodos más

utilizados según lo estudiado para esta experimentación en Abdulhamid *et al*. (2017); Chan, Yang, Yeung, & Ng (2015); Ma, Zhang, Liu, Yu, & Wang (2016); Rafique & Abulaish (2012).

3. RESULTADOS

3.1. Línea Base

En los estudios descritos en Almeida & Alberto (2013); Almeida *et al*. (2011, 2013), se indica que se ejecutaron los algoritmos descritos en WEKA con los parámetros por defecto y con un máximo de 20 iteraciones.

Tabla 3. Algoritmos evaluados.

Algoritmo	Parámetros por defecto Weka
SVML	C=1
AB	Iteraciones =10
KNN	K=1
LR	C=1
RF	Iteraciones =10

En Almeida & Alberto (2013); Almeida *et al*. (2011, 2013) no se especifica exactamente cómo se forma el patrón del token 1 y token 2. Para su ejecución en Python con la librería scikit-learn, los patrones se definieron:

- Token 1 (T1): '[^][-~][^.,:]\w+'
- Token 2:(T2): '[^ .,:\t\n\r\f\v][^ .,:\t\n\r\f\v]+'

Adicionalmente se definió evaluar los clasificadores con el patrón del token por defecto de TfidfVectorizer:

- Token por defecto (Tpd): u'(?u)\b\w\w+\b'

Tabla 4. Comparación con la línea base.

Clasificador	SC%	BH%	Acc	MCC	Clasificador
SVML+T1 (LB)	83.1	0.18	97.64	0.893	SVML+T1 (LB)
SVM L+T1	51.34	6.28	88.05	0.47	SVM L+T1
SVM L+Tpd	65.18	7.38	88.94	0.55	SVM L+Tpd
AB+T2 (LB)	84.48	0.53	97.5	0.887	AB+T2 (LB)
AB+T2	54.91	1.24	92.89	0.66	AB+T2
AB+Tpd	54.91	0.97	93.13	0.67	AB+Tpd
1NN+T2 (LB)	43.81	0.00	92.7	0.636	1NN+T2 (LB)
1NN+T2	64.73	0.00	95.28	0.78	1NN+T2
1NN+Tpd	65.62	0.00	95.40	0.79	1NN+Tpd
LR+T2 (LB)	95.48	2.09	97.59	0.899	LR+T2 (LB)
LR+T2	73.66	0.28	96.23	0.83	LR+T2
LR+Tpd	78.12	0.14	96.95	0.86	LR+Tpd
RF+T2 (LB)	65.23	0.12	95.36	0.782	RF+T2 (LB)
RF+T2	78.57	0.00	97.13	0.87	RF+T2
RF+Tpd	81.25	0.00	97.49	0.89	RF+Tpd

Los resultados de cada clasificador se comparan en la Tabla 4, con la línea base (LB) de cada uno de ellos. Se observa en los resultados, que los datos tienen variaciones respecto a la línea base y a la experimentación realizada. En SVML, AB y LR, los valores de MCC son menores en el experimento que con la línea base, pero en 1NN y RF son mayores. Esta variación se atribuye a que el patrón del token1 y del token2, muy probablemente no corresponde a los que se utilizaron como línea base, debido a que no se tiene la definición exacta del token. También se observa que en todos los casos los resultados con el token por defecto (td) para MCC son mayores al de token 1 o token 2

correspondiente, aunque solo da mejores valores que la línea base para 1NN y RF respecto al resto de métricas.

3.2. Evaluación del rendimiento

Luego de ejecutar la búsqueda por cuadrícula (grid search), se obtuvieron los resultados de la Tabla 5.

Tabla 5. Resultados con mejores parámetros.

Modelo	Parámetros	SC%	BH%	Acc%	MCC
LR	'C':777841.1071	92.41	0.21	98.80	0.95
SVML	'C': 10	91.07	0.14	98.68	0.94
SVM sigmoidal	'C': 10, 'degree': 2 'gamma': 0.3	91.07	0.28	98.57	0.94
RF	‘n_estimators': 50	82.14	0.00	97.61	0.89
AB	'n_estimators': 200	85.71	0.62	97.55	0.89
KNN	'n_neighbors': 21	70.98	0.00	96.11	0.82

Aunque la Regresión Logística obtuvo un MCC ligeramente mejor y capturó más Spam que SVM Lineal, ha bloqueado el 0.21% de mensajes legítimos, frente a sólo 0.14% de la SVM Lineal. En consecuencia, como en el filtrado de spam, un falso positivo es un error peor que un falso negativo, podemos concluir que la SVM superó a los otros métodos evaluados.

3.3. Comparación

La Tabla 6 muestra los resultados de la línea base y los resultados de clasificadores con hiper parámetros ajustados. Se puede observar que en todos los casos se logró obtener mejores valores para la métrica de calidad MCC.

Tabla 6. Comparación línea base con resultados de mejores parámetros.

Modelo	Parámetros	SC%	BH%	Acc%	MCC
LR	'C':777841.1071	92.41	0.21	98.80	0.95
LR (LB)	‘C: 1’	95.48	2.09	97.59	0.899
SVML	'C': 10	91.07	0.14	98.68	0.94
SVML (LB)	‘C: 1’	83.1	0.18	97.64	0.893
RF	‘n_estimators': 50	82.14	0.00	97.61	0.89
RF (LB)	'n_estimators': 10	65.23	0.12	95.36	0.782
AB	'n_estimators': 200	85.71	0.62	97.55	0.89
AB (LB)	'n_estimators': 10	84.48	0.53	97.5	0.887
KNN	'n_neighbors': 21	70.98	0.00	96.11	0.82
KNN (LB)	'n_neighbors': 1	43.81	0.00	92.7	0.636

4. DISCUSIÓN

En la Tabla 1, se han presentado los resultados obtenidos en los trabajos publicados por Almeida *et al*., (2011, 2013), los cuales fueron desarrollados usando WEKA. Para el presente estudio, el mismo conjunto de datos fue trabajado utilizado sentencias ejecutadas en PHYTON y de acuerdo con las medidas de calidad seleccionadas, la más relevante es el MCC, debido a que los tamaños de las clases spam y ham son muy diferentes, 13.4% de spam y 86.4% de ham. En conjunto con ella, un buen valor de SC es el que se acerca a 100% (mejor % de spam correctamente clasificados) y un buen valor de BH (menor % de ham clasificados como spam) es el que se acerca a 0%.

En cambio, al analizar en forma independiente Acc (Accuracy, Exactitud), se observa que no es una buena medida individual porque junta los valores de spam y ham correctamente clasificados entre el total del conjunto de datos y podría ocasionar que esta suma oculte la clasificación incorrecta de spam, ya que la proporción de mensajes es pequeña. Por ejemplo, para AdaBoost con n-estimators=10

para el Token 1, se obtiene Acc de 93.25% pero un SC de tan solo 57.59%, debido a que solo 129 mensajes spam fueron correctamente clasificados de un total de 225. Pero, cuando se analizan en conjunto con MCC, SH y BH, sí es posible identificar un buen clasificador.

Para los clasificadores optimizados, descritos en la Tabla 5, los tres primeros: Regresión Logística, SVM Lineal y SVM sigmoidal tienen un MCC mayor a 0.9, SC>91%, BH<0.14% y Acc>98.57%; demostrando que tienen una performance similar, con una diferencia estadística bastante pequeña. Los valores detallados pueden ser considerados una muy buena línea base para conjuntos de datos similares de mensajes spam y ham ya que superan a los descritos en el trabajo de Almeida *et al*., (2011). Mientras que para los tres últimos de la Tabla 5, si bien no alcanzaron un MCC igual o mayor a los primeros, tienen mejor valor de BH% (0 para Random Forest y KNN) con excepción de AdaBoost, que tiene el mayor valor de los modelos (0.62). Sin embargo, se puede observar que los valores obtenidos en este experimento, en comparación con la línea base en general, superaron los valores obtenidos en MCC, BH, Acc y SC (en este último a LR fue menor en 3.07 puntos) y, respecto al valor de BH, sólo tuvo un ligero incremento de 0.09, frente a los demás que disminuyeron su valor, caso notable el de Regresión Lineal de 2.09 a 0.21.

Respecto al clasificador por Regresión Logística la sintonización se realiza respecto al parámetro C, inversa de la regularización, en función de la métrica de calidad MCC obteniéndose como resultado para C = 1 valores MCC y BH de 0.863 y 1.38% respectivamente. Al aumentar el valor de C la métrica BH se mantiene constante en un valor de 2.07% y se obtiene el máximo valor MCC de 0.95 para un valor de C muy alto de 777,841.107. En Almeida *et al*. (2013) se obtuvo un MCC de 0.899

Conforme se puede observar en la Tabla 5 el clasificador kNN tiene un porcentaje nulo de bloqueo de mensajes legítimos, pero la proporción de SPAM clasificados correctamente SC también es muy baja, tanto en los resultados de línea base publicados en Almeida *et al*.(2011), como en la replicación experimental, por lo cual kNN tiene la última ubicación en cuanto a eficiencia. En el trabajo de Firte, Lemnaru, & Potolea (2010) se escribe una aplicación exclusiva de kNN para detección de SPAM con diferentes resultados.

En la Tabla 6 puede apreciarse un resumen de todos los resultados obtenidos en el presente trabajo en donde a la vez se comparan los valores reportados en los trabajos de Almeida *et al*. (2011, 2013) tomados como línea base.

5. CONCLUSIONES Y TRABAJOS FUTUROS

Se coincide con Almeida & Alberto (2013) en que el filtrado de SPAM en SMS sigue siendo un desafío, debido, principalmente, a que los mensajes de texto contienen muchas abreviaturas y modismos los cuales son susceptibles de cambio en el tiempo.

Los resultados de la experimentación en cuanto a la evaluación del rendimiento de los algoritmos aplicados se muestran en la Tabla 5, en la cual se puede apreciar que, con el conjunto de datos utilizado para simular la detección de SPAM en mensajes de textos de tipo SMS, el algoritmo de aprendizaje automático Regresión Logística es el que mejor precisión se ha alcanzado con una exactitud de 98.80%, seguido del clasificador SVM Lineal, con una precisión del 98.68%. Sin embargo, la regresión logística ha bloqueado más mensajes legítimos que SVM, por lo cual SVM podría ser considerado como el algoritmo con mejores resultados.

Comparando con trabajos previos, la presente experimentación confirma a SVM como uno de los clasificadores con mejores resultados para la detección de mensajes no deseados en SMS.

Para la experimentación descrita en este artículo en la Tokenización se partió con una semilla escrita en el código que permitió el trabajo con 2 tipos de tokens, y, conforme se ha podido observar, estos afectan a las medidas de calidad obtenidas en los diversos clasificadores, por lo cual se plantea para un trabajo futuro revisar la posibilidad de construir un algoritmo que permita una generación más amplia de tokens.

La bibliografía revisada indica que la mayoría de las investigaciones para construir clasificadores de spam SMS se basan en SVM y la red bayesiana. Tomando como referencia el estudio de

Abdulhamid *et al.*, (2017), se recomienda el uso de algoritmos biológicamente inspirados en la creación de clasificadores de spam de SMS. Estos se pueden utilizar de tres maneras distintas cuando se combinan, por ejemplo, con SVM: optimizando los pesos SVM, determinando automáticamente la estructura SVM y sus parámetros internos sin los esfuerzos del diseñador SVM y, finalmente, adaptando las reglas de aprendizaje SVM. Por esta razón se recomienda que en un trabajo futuro se debe utilizar algoritmos evolutivos en la construcción de clasificadores para la detección y clasificación de spam SMS. También se recomienda el uso de la red bayesiana en combinación con otros clasificadores para obtener un mejor rendimiento.

Finalmente, para continuar verificando la validez de estos clasificadores optimizados, se podría ejecutar sobre otros conjuntos de datos, tanto de SMS como de mensajes spam de correos, por ejemplo. También sería importante hacer una comparación de cada clasificador en cuanto a nivel resultados y tiempo de procesamiento, en este experimento, a pesar de no haber profundizado en este aspecto, la optimización del clasificador SVM no lineal fue la que demoró más tiempo en ejecutarse.

AGRADECIMIENTOS

El autor agradece la especial ayuda de Miguel Chicchón, Alfredo Oré, Silvia Vargas. Estudiantes de la Maestría en Informática de la Pontificia Universidad Católica del Perú, por la colaboración en la codificación de los algoritmos en lenguaje Phyton.

REFERENCIAS

Abdulhamid, S. M., Abd Latiff, M. S., Chiroma, H., Osho, O., Abdul-Salaam, G., Abubakar, A. I., Herawan, T. (2017). A Review on Mobile SMS Spam Filtering Techniques. *IEEE Access*, *5*, 15650-15666. https://doi.org/10.1109/ACCESS.2017.2666785

Almeida, T. A., Alberto, T. C. (2013). *Learning to block undesired comments in the blogosphere*. 12th International Conference on Machine Learning and Applications (pp. 261-266). IEEE. https://doi.org/10.1109/ICMLA.2013.133

Almeida, T. A., Gómez Hildago, J. M., Yamakami, A. (2011). Contributions to the study of SMS spam filtering. Proceedings of the 11th ACM symposium on Document engineering - DocEng ’11, pp. 259-262. https://doi.org/10.1145/2034691.2034742

Almeida, T. A., Gómez Hidalgo, J. M., Silva, T. P. (2013). Towards SMS Spam Filtering: Results under a New Dataset. *International Journal Of Information Security Science*, *2*(1), 1-18.

Bin, Z., Gang, Z., Yunbo, F., Xiaolu, Z., Weiqiang, J., Jing, D., Jiafeng, G. (2016). *Behavior analysis based SMS spammer detection in mobile communication networks*. First International Conference on Data Science in Cyberspace (DSC) (pp. 538-543). IEEE. https://doi.org/10.1109/DSC.2016.48

Chan, P. P. K., Yang, C., Yeung, D. S., Ng, W. W. Y. (2015). Spam filtering for short messages in adversarial environment. *Neurocomputing*, *155*, 167-176. https://doi.org/10.1016/J.NEUCOM.2014.12.034

Firte, L., Lemnaru, C., Potolea, R. (2010). *Spam detection filter using KNN algorithm and resampling*. Proceedings of the 2010 IEEE 6th International Conference on Intelligent Computer Communication and Processing. IEEE. https://doi.org/10.1109/ICCP.2010.5606466

Guzella, T. S., Caminhas, W. M. (2009). A review of machine learning approaches to Spam filtering. *Expert Systems with Applications*, *36*(7), 10206-10222. https://doi.org/10.1016/J.ESWA.2009.02.037

Ma, J., Zhang, Y., Liu, J., Yu, K., Wang, X. (2016). *Intelligent SMS Spam Filtering Using Topic Model*. International Conference on Intelligent Networking and Collaborative Systems (INCoS)

(pp. 380-383). IEEE. https://doi.org/10.1109/INCoS.2016.47

Rafique, M. Z., Abulaish, M. (2012). *Graph-based learning model for detection of SMS spam on smart phones*. 8th International Wireless Communications and Mobile Computing Conference (IWCMC) (pp. 1046-1051). IEEE. https://doi.org/10.1109/IWCMC.2012.6314350

Shirani-Mehr, H. (2013). *SMS Spam Detection using Machine Learning Approach*. Retrieved from https://pdfs.semanticscholar.org/a083/c8ea8e898269927e1cc0a935477175179b58.pdf

Yu, Y., Chen, Y. (2012). *A novel content based and social network aided online spam short message filter*. Proceedings of the 10th World Congress on Intelligent Control and Automation (pp. 444-449). IEEE. https://doi.org/10.1109/WCICA.2012.6357916

Zhang, X., Xiong, G., Hu, Y., Zhu, F., Dong, X., Nyberg, T. R. (2016). *A method of SMS spam filtering based on AdaBoost algorithm*. 12th World Congress on Intelligent Control and Automation (WCICA) (pp. 2328-2332). IEEE. https://doi.org/10.1109/WCICA.2016.7578522

Fuentes de Internet.

Estudio privado GSM sobre SPAM en dispositivos móviles disponible en https://www.itu.int/en/ITU-T/Workshops-and-Seminars/spam/201310/Documents/S4P1_Matias_Fernandez_Diaz.pdf