

## ALGORITMA SUPPORT VECTOR MACHINE UNTUK MENDETEKSI SMS SPAM BERBAHASA INDONESIA

**Candra Putra Susanto dan Esther Irawati Setiawan**

Teknologi Informasi

Sekolah Tinggi Teknik Surabaya

Candra.ptr19@gmail.com dan esther@stts.edu

### ABSTRAK

Pada zaman yang modern saat ini, SMS atau Short Message Service merupakan hal yang sudah dikenal oleh banyak masyarakat di dunia. Dengan sms, komunikasi antara mobile device dapat dilakukan kapan saja, di mana saja dan oleh siapa saja. Karena sifatnya yang umum, maka sms seringkali digunakan sebagai perantara untuk menyebarkan spam sesuai dengan kepentingan masing-masing individu khususnya di negara Indonesia.

*Sms Spam* adalah salah satu masalah yang menyebabkan pengguna mobile device terganggu dalam melakukan aktifitas sehari-hari. Sms Spam dapat berupa iklan-iklan televisi, tawaran-tawaran diskon belanja, dan launching produk baru. Setiap orang memiliki pandangan yang berbeda dalam mengartikan Sms Spam. Sms yang berupa iklan produk tertentu bisa saja menjadi Sms yang penting bagi beberapa pengguna mobile device. Oleh karena itu dibutuhkan algoritma yang dapat menyesuaikan setiap kondisi sesuai dengan kebiasaan pengguna. Algoritma tersebut adalah Support Vector Machine (SVM) yang termasuk dalam supervised learning yang akan mempelajari setiap kondisi yang diberikan pada saat proses pelatihan dan hasil dari pelatihan tersebut akan digunakan sebagai acuan untuk mendeteksi Sms Spam. Untuk membantu proses pelatihan pada SVM maka diperlukan algoritma Sequential Minimal Optimization (SMO) untuk mempercepat proses pelatihan.

Uji coba dilakukan dengan menggunakan metode cross validation testing memberikan akurasi kebenaran sebesar 88% dan diuji dengan 100 Sms berbahasa Indonesia sebagai uji coba yang menghasilkan presentase 79%. Berdasarkan hasil uji coba, menunjukkan bahwa dengan 155 sms untuk pelatihan dapat menangani 100 sms dari pengguna yang berbeda dengan gaya penulisan yang berbeda.

Kata kunci: *Support Vector Machine, Spam Filtering, Short Message Service*

### ABSTRACT

*In today's modern era, SMS or Short Message Service is already known by many people in the world. With sms, the communication between mobile devices can be done anytime, anywhere, and by anyone. Because the nature of sms is public, therefore sms is often used to spread spam according to the personal goal of every individual especially in Indonesia.*

*Sms spamming is one of the problems that can be annoying the mobile device users while doing their daily activities. Sms Spamming may include television advertising, shopping discount offer, and promotions new products. Everyone has a different view to*

*Sms Spamming. Sms that include the advertising for particular product may be an important sms for some users of mobile devices. Therefore, it is required that the algorithm can adjust every condition according to the user's habits. The algorithm is a Support Vector Machine (SVM) which is categorized as the supervised learning that will study each condition given during the training process and the results will be used as a knowledge for detecting sms spamming. In order to help the SVM training process, Sequential Minimal Optimization (SMO) is needed to improve the speed of the process.*

*The test is done by using cross validation testing method that gives an accuracy of 88% right and tested with 100 Indonesian language sms which is showed the percentage of 79%. Based on the test results, provided 155 sms as dataset for training process, this system can handle 100 sms from different users with different writing styles.*

*Keywords: Support Vector Machine, Spam Filtering, Short Message Service*

## **I. PENDAHULUAN**

Pengguna mobile device yang menerima sms spam tidak perlu khawatir merasa terganggu karena dengan menggunakan algoritma SVM maka sms-sms yang terdeteksi sebagai sms spam dapat dikelompokkan tersendiri sehingga pengguna mobile device tinggal menyetujuinya sebagai sms spam atau tidak. Kemudian sistem akan berjalan secara otomatis. Dengan pengelompokan tersebut, pengguna dapat mengatur pesan dengan mudah. Jika terdapat kesalahan dalam mendeteksi sms spam, maka pengguna dapat memberikan tanggapan dan masukan terhadap sistem sehingga algoritma SVM dapat "belajar" lebih baik dan lebih baik lagi.

Masukan atau feedback yang diberikan pengguna pada sistem tentunya tidak serta merta dapat langsung dijadikan referensi dalam melakukan pelatihan terhadap algoritma SVM tetapi diverifikasi terlebih dahulu kebenarannya. Verifikasi tersebut memerlukan bantuan pihak tertentu yang berinteraksi dengan sistem secara langsung. Contohnya jika terdapat administrator yang secara online dapat melakukan verifikasi hal tersebut. Seperti yang telah diketahui bahwa algoritma SVM termasuk dalam disiplin ilmu machine learning yang akan mempelajari setiap bentuk dan pola yang diajarkan dan akan memprediksi setiap sms sesuai dengan kelas yang diberikan yaitu spam atau bukan spam. Dengan sifatnya yang fleksibel, SVM dapat menyesuaikan diri dengan tingkah laku dan kebiasaan pengguna dalam menggunakan sms sebagai media komunikasi sosial.

Kenyamanan pengguna mobile device dapat tercapai jika hal-hal yang tidak diinginkan dapat dihindari. Kebanyakan penanggulangan spam hanya berfokus pada email atau pengguna internet. Hal ini yang menyebabkan aplikasi yang digunakan untuk melakukan pendeteksian spam pada mobile device tidak sebanyak yang sudah ada pada pengguna internet khususnya pengguna email. Oleh karena itu paper ini bertujuan untuk mengimplementasikan pendeteksian sms spam pada mobile device mengingat populernya pengguna mobile device saat ini. Seperti yang diketahui bahwa mobile device saat ini sangat populer berkat keunggulan sistem operasi Android, iOS, dan Windows yang menyajikan tampilan yang interaktif dengan didukung spesifikasi mobile device yang canggih. Dengan populernya ketiga sistem operasi tersebut menandakan bahwa pengguna mobile device sudah semakin meningkat dan bahkan melebihi pengguna Personal Computer (PC). Dengan semakin berkembangnya teknologi, tidak menutup kemungkinan bahwa mobile device dapat menggantikan fungsi-fungsi penting pada PC.

## II. SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) diciptakan oleh Vladimir N. Vapnik dan kemudian dilakukan perkembangan oleh Vapnik dengan temannya Corinna Cortes pada tahun 1995. Dalam machine learning, support vector machine termasuk dalam model supervised learning atau pembelajaran dengan pengawasan yang berhubungan dengan analisis data dan pengenalan pola. SVM juga dapat digunakan untuk klasifikasi dan analisis regresi. Metode dasar SVM adalah dengan mengambil satu set input data dan memperkirakan untuk setiap input yang diberikan dari dua kelas yang memungkinkan untuk membentuk sebuah output. Oleh karena input yang diberikan adalah dua kelas maka SVM dapat disebut sebagai biner classifier.

Algoritma pelatihan SVM membangun sebuah model yang memberikan pola-pola sample baru dalam satu kategori atau lebih. Model SVM merupakan representasi dari sample sebagai titik dalam ruang yang dipetakan dalam dimensi yang tinggi sehingga sample dari kategori tersebut dibagi dengan jarak yang seluas mungkin. Sample baru kemudian dipetakan ke dalam ruang yang sama dan diperkirakan termasuk kategori berdasarkan sisi mana dari tiap class yang cenderung memiliki kemiripan dengan kelas yang diberikan. SVM memiliki dua jenis klasifikasi yaitu klasifikasi linear dan klasifikasi non linear menggunakan trik kernel yang secara implisit memetakan inputan ke dalam ruang fitur berdimensi tinggi.

SVM merupakan supervised machine learning dimana membutuhkan proses learning terlebih dahulu sebelum melakukan uji coba. SVM dimulai dengan membentuk sebuah hyperplane untuk setiap data yang digunakan untuk proses pelatihan. Hyperplane yang paling baik diperoleh dengan memaksimalkan margin atau jarak. Nantinya sample yang paling mirip dan dekat dengan hyperplane dapat disebut sebagai support vector. Proses pembelajaran pada SVM adalah dengan mencari support vector untuk mencari hyperplane yang terbaik. Untuk memperoleh hyperplane yang terbaik maka diperlukan perhitungan yang dapat menentukan jarak antara training set dengan hyperplane yang ada. Pada proses pembelajaran pertama kali akan dicari jarak minimum dari setiap training set yang ada terhadap hyperplane dengan menggunakan rumus bentuk primal di bawah ini.

$$|\vec{w}|^2$$

$$y_i(\langle \vec{w}, \vec{x}_i \rangle) + b \geq 1 \quad (i = 1, 2, \dots, l) \dots \dots \dots (1)$$

Bentuk primal adalah bentuk utama yang ditemukan saat melakukan proses pelatihan pada SVM. Sedangkan bentuk dual merupakan bentuk ganda yang digunakan untuk memaksimalkan nilai alpha. Dengan memaksimalkan nilai alpha maka diharapkan akurasi yang dihasilkan dalam memprediksi sms spam menjadi lebih baik. Bentuk dual seperti terlihat pada rumus berikut ini.

$$\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j \langle \vec{x}_i, \vec{x}_j \rangle$$

$$\alpha_i \geq 0 \quad (i = 1, 2, \dots, l) \quad \sum_{i=1}^l \alpha_i y_i = 0 \dots \dots \dots (2)$$

Pada rumus 1 dan 2 merupakan masalah quadratic programming yang dimana alpha yang didapatkan bernilai 0. Data dari proses pelatihan yang alphanya bernilai lebih besar dari 0 disebut sebagai support vector. Support vector tersebut yang menjadi knowledge atau pengetahuan dalam mengidentifikasi sms spam

Dalam melakukan proses pembelajaran, SVM mengalami beberapa kendala dalam hal waktu. Karena banyaknya langkah-langkah yang dilakukan maka membuat proses pembelajaran tidak berjalan secara efisien. Untuk itu dibutuhkan algoritma lain yang digunakan untuk membantu menyelesaikan proses pembelajaran pada SVM secara efisien dan cepat. Algoritma tersebut adalah Sequential Minimal Optimization (SMO).

Pada proses pembelajaran atau pelatihan yang menjadi tujuan utama adalah menemukan nilai alpha. Jika alpha ditemukan maka dapat ditentukan nilai bobot dari tiap sms yang dijadikan bahan pembelajaran. Ketika bobot tersebut ditemukan maka dapat diperoleh pola-pola yang dibutuhkan untuk melakukan klasifikasi dengan menggunakan SVM. Pola-pola tersebut yang berguna untuk menentukan klasifikasi dari dua kelas yang telah ditentukan pada kasus sms spam yaitu kelas spam atau kelas non spam. Data dari hasil uji coba akan dihitung dengan fungsi keputusan dan dikalikan dengan alpha dan kelas dari data pelatihan yang ada. Seperti terlihat pada rumus di bawah ini.

$$f(\vec{t}) = \text{sgn}[\sum_{i=1, x_i \in SV}^n \alpha_i y_i < \vec{t}, \vec{x}_i > + b] \dots\dots\dots (3)$$

Dimana  $\alpha_i$  adalah alpha ke-i,  $y_i$  adalah kelas dari data pelatihan yang ke-i,  $\vec{t}$  adalah vektor dari data uji coba,  $\vec{x}_i$  adalah vektor dari data pelatihan dan  $b$  adalah bias.

Pada Rumus 1 di atas merupakan fungsi keputusan atau decision function untuk permasalahan linear SVM. Sedangkan dalam masalah non linear, tidak jauh berbeda rumus yang digunakan hanya perbedaannya adalah adanya trik kernel atau metode kernel untuk membentuk titik-titik untuk setiap vektor pada data pelatihan. Fungsi keputusan ditambahkan fungsi kernel untuk mencari perbandingan berpasangan untuk tiap kelas yang diberikan dan kemudian ditambahkan dengan nilai bias yang didapatkan.

Dimana  $\alpha_i$  adalah alpha ke-i,  $y_i$  adalah kelas dari data pelatihan ke-i,  $\vec{t}$  adalah vektor dari data uji coba,  $\vec{x}_i$  adalah vektor dari data pelatihan dan  $b$  adalah bias. Hasil yang diperoleh pada fungsi keputusan seperti terlihat pada rumus 2 adalah sebuah angka positif 1 atau negatif 1. Angka 1 merupakan kelas pertama yaitu spam. Sedangkan angka -1 merupakan kelas kedua atau kelas non spam. Untuk simbol sgn pada rumus 2 merupakan singkatan dari sign yang berarti setiap nilai dari fungsi keputusan akan menghasilkan dua nilai yang berbeda (1 atau -1) yang menunjukkan kelas yang berbeda.

$$f(\vec{t}) = \text{sgn}[\sum_{i=1, x_i \in SV}^n \alpha_i y_i K(\vec{t}, \vec{x}_i) + b] \dots\dots\dots (4)$$

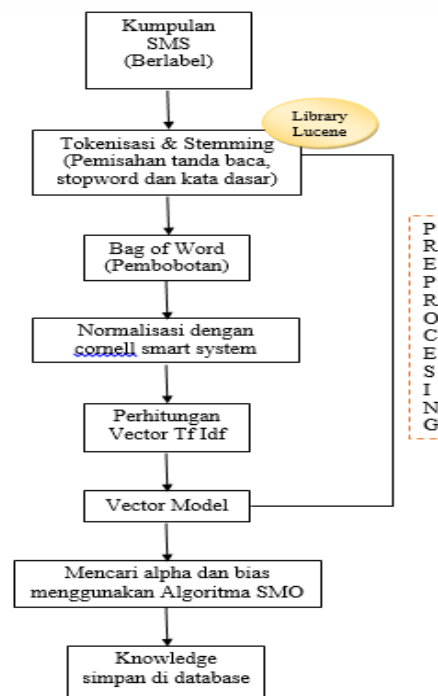
Dalam melakukan proses klasifikasi akan dilakukan perhitungan dari tiap support vector yang ada pada basis data. Kemudian untuk setiap vector pada pelatihan akan dibandingkan dengan vector pada uji coba. Perbandingan tersebut menggunakan bantuan metode kernel gaussian. Kernel Gaussian merupakan sebuah metode untuk mendapatkan nilai perbandingan berpasangan dari dua buah vektor. Perbandingan berpasangan tersebut bertujuan untuk mengetahui seberapa mirip dari kedua vector yang diberikan. Rumus kernel Gaussian seperti pada rumus berikut ini.

$$K(\vec{x}, \vec{y}) = \exp\left[-\frac{\|\vec{x} - \vec{y}\|^2}{2\sigma^2}\right] \dots\dots\dots (5)$$

Dimana  $\vec{x}$  adalah vector pertama dan  $\vec{y}$  adalah vector kedua sedangkan  $\sigma$  adalah gamma. Konstanta gamma merupakan ketetapan yang ditentukan sebelum melakukan perhitungan. Ketetapan tersebut biasanya berkaitan dengan hasil penelitian tertentu yang mendukung dalam menentukan nilai konstanta yang digunakan dalam perhitungan. Semakin baik nilai konstanta yang diberikan maka akan semakin memaksimalkan hasil prediksi yang dihasilkan.

### III. UJI COBA PENDETEKSIAN SMS SPAM

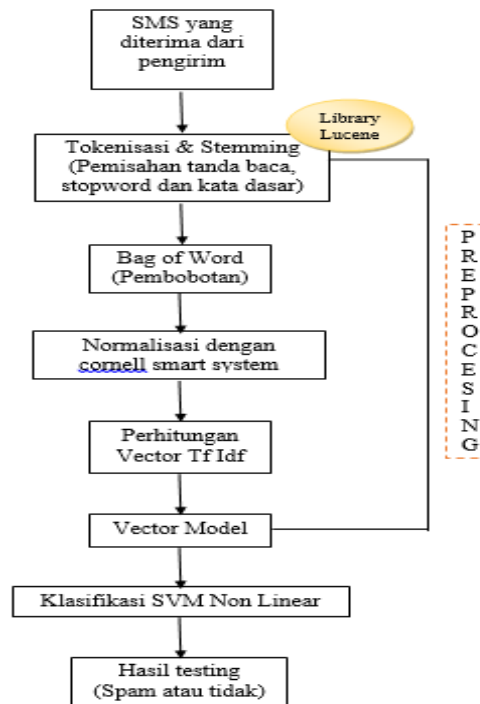
Dalam prosesnya algoritma support vector machine dibagi menjadi 2 fase yaitu fase pelatihan dan fase uji coba. Fase pelatihan dan uji coba dalam prosesnya akan dibantu dengan library lucene dari Apache yang sudah menyediakan proses tokenisasi dan stemming dalam bahasa Indonesia. Setelah proses tokenisasi dan stemming selesai kemudian dilanjutkan dengan proses pembobotan untuk tiap kata unik yang ada. Setelah itu nilai bobot dinormalisasi dan dilakukan perhitungan tf.idf. Kemudian terbentuklah vektor model yang nantinya digunakan dalam proses pelatihan. Proses pelatihan SVM akan dibantu dengan algoritma SMO. Hasil pelatihan tersebut adalah nilai alpha dan bias. Nilai alpha dan bias tersebutlah yang merupakan knowledge yang digunakan dalam proses uji coba. Knowledge tersebut disimpan dalam basis data dengan tujuan dapat diakses kapan saja. Untuk lebih jelasnya fase pelatihan seperti pada gambar berikut ini.



**Gambar 1. Fase Pelatihan**

Pada fase uji coba memiliki langkah awal atau preprocessing yang sama dengan fase pelatihan yaitu tokenisasi stemming, bag of word, normalisasi, perhitungan tf.idf dan pembentukan vektor model. Perbedaan antar fase pelatihan dengan fase uji coba adalah

pada langkah terakhir dimana fase pelatihan melakukan penyimpanan knowledge sedangkan fase uji coba melakukan klasifikasi pada setiap pesan. Fase uji coba akan menghasilkan sebuah keputusan yang digunakan untuk memprediksi suatu pesan pada kelas tertentu. Hasil keputusan tersebut merupakan hasil dari perbandingan dengan knowledge yang dihasilkan pada fase pelatihan. Berikut ini adalah garis besar langkah-langkah yang dilakukan pada fase uji coba.



**Gambar 2. Fase Uji Coba**

Dari fase uji coba akan dihasilkan keputusan yang menyatakan sms termasuk dalam kelas spam atau non spam. Seperti yang sudah dijelaskan sebelumnya bahwa hasil dari keputusan yang diberikan hanya ada dua kelas yaitu 1 atau -1 yang masing-masing merepresentasikan kelas yang berbeda. Uji coba akan dilakukan dengan metode cross validation testing 10 fold atau 10 kali perulangan seperti pada tabel berikut ini.

**Tabel 1. Hasil Uji Coba Cross Validation Testing 10 Fold**

Keterangan Jenis Pesan	Range Data Pelatihan (Berbahasa Indonesia)	Range Data Uji Coba (Berbahasa Indonesia)	Hasil Prediksi Spam (%)	Hasil Prediksi Non Spam (%)
Pesan spam	139 sampai 140 Pesan	15 sampai 16 Pesan	85.73%	14.27%
Pesan non spam			7.12%	92.88%
Total Keseluruhan			<b>88.13 %</b>	<b>11.87%</b>

Seperti yang terlihat pada tabel 1 di atas bahwa hasil uji coba dengan cross validation testing 10 fold menghasilkan prediksi yang benar dengan presentase sebesar 88.13% yang berarti bahwa performa dari SVM berjalan dengan baik. Range data pelatihan antara 139 sms sampai dengan 140 sms karena data yang digunakan untuk uji coba total keseluruhannya adalah 155 pesan sehingga jika dibagi menjadi 10 bagian sama rata akan menghasilkan 5 bagian berisi masing-masing 16 pesan dan 5 bagian sisanya berisi masing-masing 15 pesan. Uji coba selanjutnya adalah uji coba dengan 100 pesan berbahasa Indonesia seperti terlihat pada tabel berikut ini.

**Tabel 2. Hasil Uji Coba 100 Pesan Berbahasa Indonesia**

<b>Keterangan Jenis Pesan</b>	<b>Jumlah Data Pelatihan (Berbahasa Indonesia)</b>	<b>Jumlah Data Uji Coba (Berbahasa Indonesia)</b>	<b>Hasil Prediksi Spam (%)</b>	<b>Hasil Prediksi Non Spam (%)</b>
Pesan spam	103 pesan	69 pesan	72.46%	27.54%
Pesan non spam	52 pesan	31 pesan	6.45%	93.55%
Total Keseluruhan	<b>155 pesan</b>	<b>100 pesan</b>	<b>79 %</b>	<b>21%</b>

Berdasarkan hasil uji coba yang dilakukan, lamanya proses pelatihan yang dibutuhkan adalah antara 10 menit sampai dengan 18 menit bergantung pada jumlah karakter pada tiap pesan atau sms yang ada. Semakin banyak karakter maka semakin lama proses pelatihan berlangsung. Hal ini juga berlaku pada saat proses uji coba. Dalam uji coba yang dilakukan pada mobile device menunjukkan bahwa dalam setiap kali melakukan prediksi sms spam akan membutuhkan waktu paling cepat 20 detik dan paling lambat 120 detik. Untuk pesan dengan karakter yang keseluruhannya unik dapat memberikan lama waktu yang berbeda bahkan lebih lama dari waktu yang diperkirakan. Dalam proses prediksi pesan, waktu yang dibutuhkan relatif lebih cepat dibandingkan saat melakukan proses pembelajaran. Tetapi mengingat bahwa setiap mobile device memiliki spesifikasi yang berbeda maka hal ini akan menjadi masalah. Masalah tersebut timbul karena terbatasnya spesifikasi mobile device dibandingkan dengan komputer pc.

#### **IV. PENUTUP**

Dari hasil uji coba yang dilakukan diketahui bahwa tingkat akurasi dan pembelajaran SVM menunjukkan akurasi yang baik yaitu 79% saat diuji dengan 100 sms berbahasa Indonesia dan 88% saat diuji dengan metode cross validation testing. Walaupun bekerja dengan baik, pengimplementasian SVM pada mobile device tetap membutuhkan pengembangan lebih lanjut karena keterbatasan spesifikasi mobile device dalam menyelesaikan setiap proses dalam SVM mengakibatkan semakin lamanya waktu yang dibutuhkan.

Proses training SVM membutuhkan waktu yang cukup lama tergantung pada banyaknya dokumen pelatihan. Dokumen pelatihan yang banyak memberikan akurasi yang lebih tinggi meskipun prosesnya lebih lama, selain itu banyaknya kata unik yang terkandung pada setiap pesan atau sms juga mempengaruhi lamanya proses pelatihan. Berdasarkan dari hasil uji coba menunjukkan bahwa lamanya proses pelatihan SVM pada mobile device adalah 10 menit sampai dengan 18 menit. Sedangkan untuk proses uji coba

prediksi sms spam yaitu antara 20 detik sampai dengan 120 detik untuk setiap sms. Hal ini menunjukkan bahwa waktu yang dibutuhkan untuk memprediksi sebuah sms spam masih kurang cepat karena pengguna mobile device tentunya menginginkan proses yang lebih cepat untuk mendeteksi adanya sms spam. Pengembangan selanjutnya dari paper ini adalah bagaimana mengimplementasikan SVM pada mobile device dengan kinerja yang lebih efisien dan fleksibel menyesuaikan dengan spesifikasi setiap mobile device yang ada sehingga pengguna mobile device dapat menggunakannya dengan nyaman. Saran dan dukungan dari pengguna mobile device juga dapat digunakan sebagai bahan untuk memberikan tingkat akurasi yang lebih baik dan sesuai dengan kebutuhan setiap pengguna.

## V. DAFTAR PUSTAKA

- [1] Drucker, Harris, Wu, Donghui, Vapnik, N., Vladimir. *Support Vector Machine for Spam Categorization*. September. IEEE Transactions On Neural Networks vol. 10 no. 5. 1999.
- [2] Holden, S. *Spam Filters*. September. 2013. <http://www.freshmeat.net/articles/view/964>.
- [3] Satriyo, Anto , Dr. *Pengantar Support Vector Machine (SVM)*. Pusat Teknologi Informasi dan Komunikasi BPP Teknologi. 2003.
- [4] Lutz, Hamel. *Knowledge Discovery with Support Vector Machine*. John Wiley & Sons, Inc. 2009.
- [5] J.C. Platt. *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machine*. Microsoft Research. April. 1998.
- [6] *Support Vector Machine*. 2014. [http://en.wikipedia.org/wiki/Support\\_vector\\_machine](http://en.wikipedia.org/wiki/Support_vector_machine).