Detección y diagnóstico de fallas para la dinámica lateral de un automóvil utilizando máquinas de soporte vectorial multiclase

Jesús Alejandro Navarro Acosta, Juan Pablo Nieto González

Corporación Mexicana de Investigación en Materiales, S.A. de C.V. (COMIMSA), México

{jesus.navarro, juan.nieto}@comimsa.com

Resumen. En la actualidad el correcto y eficiente monitoreo de sistemas altamente complejos como el automóvil, representa una tarea desafiante para la ingeniería en general. Los objetivos primordiales en un sistema de detección y diagnóstico de fallas para un vehículo son: evitar daño en el mismo y evitar situaciones de peligro para los tripulantes. Los principales retos para lograr un óptimo monitoreo de un automóvil se derivan de la fuerte correlación entre diversas variables, lo cual aumenta la probabilidad de generar falsas alarmas. Además de la existencia de incertidumbre inherente al sistema causada por mediciones con presencia de ruido. En este artículo se presenta un sistema de monitoreo y diagnóstico de la dinámica lateral de un vehículo. El mejor desempeño de la nueva propuesta se valida comparando los resultados obtenidos contra los de un sistema de monitoreo encontrado en la revisión del estado del arte y que combina tres técnicas para dar el diagnóstico final. Tales técnicas son escalamiento multidimensional (EMD), rangos de percentiles (RP) y análisis de componentes principales (ACP). Se presenta un caso de estudio en donde el sistema de diagnóstico se diseñó para el monitoreo de seis variables. Los datos se obtuvieron mediante el empleo del simulador VEHDYNA. Se muestra como los resultados obtenidos con la nueva propuesta son prometedores.

Palabras clave: Automóvil, SVM multiclase, dinámica lateral.

1. Introducción

Debido al constante avance de la tecnología, el sector industrial es capaz de elaborar productos cada vez más complejos. Por tal motivo, llevar a cabo un correcto y eficiente monitoreo para la detección y diagnóstico de fallas es un reto para la ingeniería actual. La detección y diagnóstico de fallas en sistemas de ingeniería está relacionada con la detección de fallas en máquinas complejas mediante aprendizaje de patrones específicos de comportamiento en los datos observados. Un vehículo moderno es un ejemplo de tal sistema de ingeniería complejo en el que hay un gran número de sensores, controladores y módulos informáticos

integrados en el vehículo, los cuales se encargan de recolectar un gran número de señales [1]. Dichas señales cuentan con características como rangos variables de magnitud, oscilación, frecuencia, etc. Por tal motivo, el procesamiento de estos datos es una tarea extremadamente difícil para los sistemas basados en técnicas de control clásico. Debido a esto, es importante el estudio e implementación de técnicas de inteligencia artificial dotadas de la flexibilidad necesaria para procesar distintos tipos de variables. Las cuales tengan como común denominador la presencia de ruido, correlación, características no lineales, etc. Las máquinas de soporte vectorial (MSV) con base en la teoría del aprendizaje estadístico y principio de minimización del riesgo estructural, se han aplicado en el ámbito del reconocimiento de fallas por su excelente capacidad de generalización [2]. En este artículo se presenta un sistema de detección y diagnóstico de fallas basado en máquinas de soporte vectorial multiclase (MSVMC) el cual requiere datos en modo de operación normal y datos en modo de falla para su entrenamiento. Su desempeño es comparado frente a un sistema de detección y diagnóstico de fallas propuesto por [1], el cual combina escalamiento multidimensional (EMD), rangos de percentiles (RP) y análisis de componentes principales (ACP). Los resultados se obtienen a partir de una simulación para un modelo de la dinámica lateral de un automóvil el cual consta de seis variables. La organización del artículo es la siguiente: la sección 2 presenta brevemente los principios de MSV, MSVMC, EMD, RP y ACP. La sección 3 presenta las descripciones generales de los sistemas de monitoreo a comparar. La sección 4 muestra el caso de estudio. La sección 5 muestra como ambos sistemas trabajan frente a una simulación para fallas individuales (fallas sobre una variable a la vez). La sección 6 presenta las conclusiones.

2. Técnicas

2.1. Maquinas de soporte vectorial

Las máquinas de soporte vectorial son un algoritmo de aprendizaje automático que se utiliza para la clasificación y la regresión de conjuntos de datos de alta dimensionalidad. Este ofrece grandes resultados debido a sus capacidades para lidiar con problemas como mínimo local, pocas muestras de datos y problemas no lineales. Las MSV estándar son utilizadas para problemas de clasificación binaria [3]. La idea central de esta técnica es determinar una separación lineal (hiperplano separador) el cual es orientado en dirección tal que su distancia a los puntos de datos más cercanos en cada una de las dos clases (margen) sea la máxima. Los puntos de datos más cercanos son conocidos como vectores de soporte [4].

Caso linealmente separable. Se tienen vectores de entrada $\mathbf{x}_i \in \mathbb{R}^d$ $(i = 1, 2, ..., n)$ correspondientes con las etiquetas $\mathbf{y}_i \in \{-1, +1\}$. Existe un plano separador cuya función es [5]

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{1}$$

Donde $\mathbf{w} \in \mathbb{R}^n$ es un vector normal que define la frontera, b es un umbral escalar y $\frac{|b|}{\|\mathbf{w}\|}$ representa la distancia perpendicular desde el hiperplano separador hacia el origen. Mientras que la distancia del hiperplano al margen esta dada por $d = \frac{1}{\|\mathbf{w}\|}$. La Figura 1 muestra el hiperplano separador y el margen máximo entre clases (lineas punteadas).

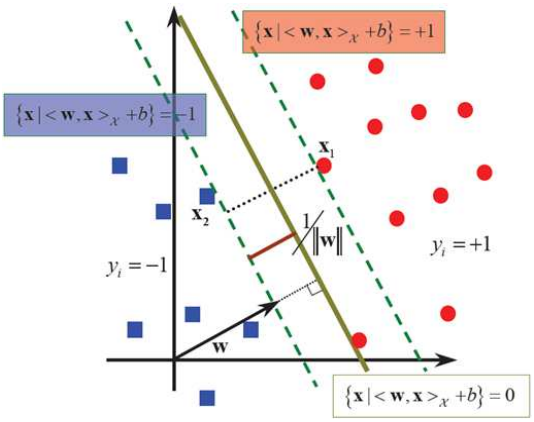

Fig. 1. Hiperplano y definición del margen geométrico, adaptado de [6].

Dos hiperplanos paralelos pueden ser representados como

$$\mathbf{y}_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) \geq 1 \tag{2}$$

Como se definió anteriormente MSV trata de maximizar la distancia entre dos clases, donde la amplitud del margen entre dos hiperplanos paralelos es $d = \frac{2}{\|\mathbf{w}\|}$, por lo tanto para el caso linealmente separable se puede encontrar el hiperplano óptimo resolviendo el siguiente problema de optimización cuadrático

$$\begin{gathered} min\ \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{sujeto a: } \mathbf{y}_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) \geq 1 \end{gathered} \tag{3}$$

Utilizando multiplicadores de Lagrange $\alpha_i (i = 1, 2, ..., n)$ para la restricción, el problema primal (Ecuación (4)) se convierte en encontrar el punto de silla de Lagrange. Por lo tanto se vuelve un problema dual de la forma

$$max\ L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{n}^{ij} \alpha_i \alpha_j \mathbf{y}_i \mathbf{y}_j$$
$$\text{sujeto a: } \sum_{i=1}^{n} \alpha_i \mathbf{y}_i = 0,\ \alpha_i \geq 0 \tag{4}$$

Aplicando las condiciones de Karush-Kukn-Tucker (KKT), se tiene

$$\alpha_i \left[\mathbf{y}_i \left(\mathbf{w} \cdot \mathbf{x}_i + b \right) - 1 \right] = 0 \tag{5}$$

Si $\alpha_i > 0$, los puntos correspondientes a los datos son llamados vectores de soporte. Por lo tanto, la solución óptima para el vector normal está dada por

$$\mathbf{w}^* = \sum_{i=1}^{N} \alpha_i \mathbf{y}_i \mathbf{x}_i \tag{6}$$

Donde N es el número de vectores de soporte. De la ecuación (5), eligiendo cualquier vector de soporte $(\mathbf{x}_k, \mathbf{y}_k)$, se obtiene $b^* = \mathbf{y}_k - \mathbf{w}^* \cdot \mathbf{x}_k$. Después $(\mathbf{w}^*, b^*)$ es determinado. A partir de la ecuación (6) el hiperplano separador óptimo puede escribirse como [7,8]

$$f(\mathbf{x}) = \mathbf{sgn} \left(\sum_{i=1}^{N} \alpha_i \mathbf{y}_i (\mathbf{x} \cdot \mathbf{x}_i) + b^* \right)$$
$$\text{donde sgn(.) es la función sign, y } \mathbf{x} \in \begin{cases} +1 \text{ si } f(\mathbf{x}) > 0 \\ -1 \text{ si } f(\mathbf{x}) < 0 \end{cases} \tag{7}$$

Caso no linealmente separable. Las MSV también pueden manejar casos donde los datos no son linealmente separables. Estas intentan mapear el vector de entrada $\mathbf{x}_i \in \mathbb{R}^d$ sobre un espacio de mayor dimensionalidad. Este proceso está basado en la elección de una función kernel. Algunas de las funciones kernel ($\mathbf{K}$) más usadas son [9]:

- Kernel lineal
- Kernel polinomial
- Función de base radial
- Kernel sigmoideo

Por lo tanto el hiperplano óptimo (Ec. (8)) toma la forma

$$f(\mathbf{x}) = \mathbf{sgn} \left(\sum_{i=1}^{N} \alpha_i \mathbf{y}_i \cdot \mathbf{K}(\mathbf{x} \cdot \mathbf{x}_i) + b^* \right) \tag{8}$$

La Figura 2 muestra un ejemplo de una transformación no lineal

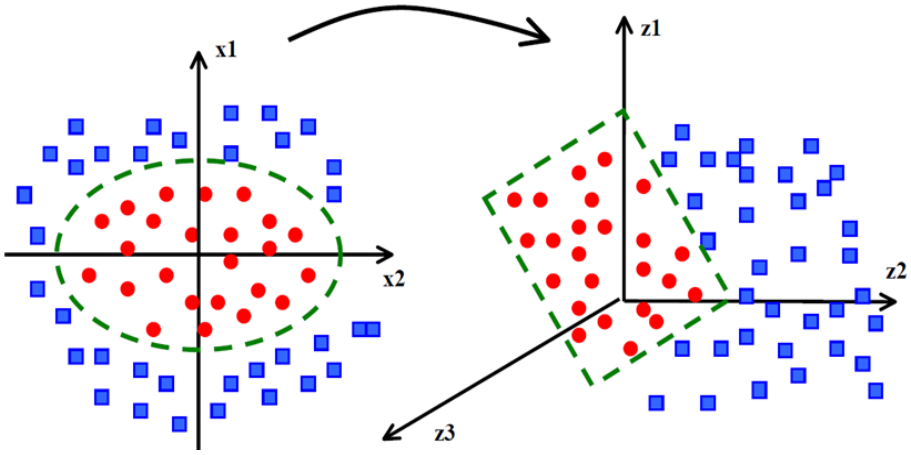

Fig. 2. Transformación no lineal, adaptada de [6].

2.2. Escalamiento multidimensional (EMD)

La técnica EMD es aplicada cuando para un conjunto de distancias observadas entre cada par de N objetos, se desea encontrar una representación de los mismos en dimensiones menores, tal que las proximidades entre objetos coincidan lo más cerca posible con las originales [1].

Para N objetos obtenemos

$$M = \frac{N(N-1)}{2} \tag{9}$$

Se ordenan las semejanzas $s_{i_1k_1} < s_{i_2k_2} < ... < s_{i_Mk_M}$ donde $s_{i_1k_1}$ es la más pequeña de las M semejanzas. Ahora se calculan las distancias entre objetos $d^{(q)}_{i_1k_1} > d^{(q)}_{i_2k_2} > ... > d^{(q)}_{i_Mk_M}$. Por último se minimiza el *stress*

$$stress = \left[\frac{\sum_{i<k} \sum \left(d^{(q)}_{ik} - \widehat{d}^{(q)}_{ik} \right)^2}{\sum_{i<k} \sum (d^{(q)}_{ik})^2} \right]^{\frac{1}{2}} \tag{10}$$

Usando las $\widehat{d}^{(q)}_{ik}$ mueve los puntos alrededor para obtener una mejor configuración, el proceso se repite hasta que la mejor representación es obtenida (*stress* mínimo).

2.3. Rangos percentiles

Los percentiles se aplican a datos numéricos para describir la dispersión en la muestra de datos. Los percentiles se basan en la división en 100 unidades.

El P-ésimo divide los datos de manera que existe un P porcentaje de números por debajo del P-ésimo percentil y un porcentaje igual a $100 - P$ por encima del mismo. Los percentiles se calculan ordenando las observaciones en orden descendente, después calcule las posición:

$$L = \frac{P}{100} * N \tag{11}$$

donde N es el número total de observaciones.

Rangos percentiles calculan la diferencia entre dos percentiles de una muestra N y ya que estos brindan información acerca de la dispersión y localización de los datos, esta diferencia es una forma robusta de estimar la dispersión de los datos aun cuando estos no se distribuyan normalmente [1].

2.4. Análisis de componentes principales

Considere una matriz de datos $\mathbf{X}$ de tamaño $m \times n$, para m variables del proceso con n mediciones de cada variable. Asumimos que los datos para cada variable han sido escalados con media 0 y varianza 1. La matriz de covarianza de $\mathbf{X}$ esta definida como

$$\sum = \frac{\mathbf{X}^T \cdot \mathbf{X}}{n-1} \tag{12}$$

Sea $\lambda_i (i = 1, 2, \ldots, m)$ los valores propios de la matriz de covarianza, los cuales son organizados en orden descendente para determinar los componentes principales (CPs), y los componentes principales $\mathbf{p}_i$ (*loadings*) los cuales son los correspondientes vectores propios. Entonces, los primeros K CPs son seleccionados para construir el modelo ACP, por lo que la matriz de datos $\mathbf{X}$ puede ser expandida usando los componentes principales $\mathbf{p}_i$, los vectores $\mathbf{t}_i$ (*scores*), y la matriz residual $\mathbf{E}$ [10]

$$\mathbf{X} = \sum_{i=1}^{K} \mathbf{t}_i \mathbf{p}_i + \mathbf{E} \tag{13}$$

Los $\mathbf{t}_i$ contienen información sobre como las muestras estan relacionadas entre si. Mientras los $\mathbf{p}_i$ contienen información sobre la relación existente entre las variables. El número de componentes principales debe ser igual o menor que las variables de $\mathbf{X}$ [1].

2.5. Matriz de confusión

En el área de la inteligencia artificial la matriz de confusión es una herramienta de visualización que se usa en el aprendizaje supervisado. Cada columna de la matriz representa el número de predicciones de cada clase, y cada fila representa cada clase real. La matriz de confusión facilita observar si el sistema confunde dos clases.

3. Descripción del sistema

Como se mencionó anteriormente el desempeño del sistema de monitoreo propuesto en este artículo será comparado con el sistema fuera de línea propuesto por [1]. A continuación se presenta una breve descripción del funcionamiento de dicho sistema.

El sistema puede ser clasificado como un método basado en datos históricos del proceso, este necesita conjuntos de datos del comportamiento del sistema en estado normal de operación. La etapa de extracción de características se lleva a cabo para que el sistema aprenda el estado normal de operación del proceso. Primero calcula ACP para observar la correlación entre variables. Esto agrupará las variables que estén correlacionadas y separará las que no lo estén. Esto puede revelar la causa raíz de la falla. Después lleva a cabo EMD para localizar variables con correlación positiva y variables con correlación negativa. Estas distancias son utilizadas para obtener límites en los cuales se encuentran los datos de operación normal. De esta manera se localiza la falla cuando esta suceda. Por último calcula RP para obtener un valor que describa la distribución de los datos. Una falla puede ser detectada comparando los RP de los datos de prueba contra los datos de operación normal, si estos son similares se asume que no existe falla de lo contrario se calcula nuevamente ACP para observar las variables que están en falla. Finalmente para localizar la falla se observan las posiciones de las distancias en el vector de salida del EMD, las cuales se encuentren fuera de los limites obtenidos en la fase de aprendizaje [1].

Una vez descrito el funcionamiento de este sistema, se describe el propuesto en el presente artículo. El sistema de monitoreo basado en Máquinas de Soporte Vectorial necesita bases de datos tanto en modo de operación normal como en modo de falla para el aprendizaje. Como se muestra en la Figura (3) el sistema no cuenta con ningún tipo de pre procesamiento. Una vez generadas las bases de datos estas sirven como entrada para el entrenamiento y validación del sistema. Si el desempeño del sistema es adecuado este se encuentra listo para recibir datos de prueba. A continuación se muestra un resumen del funcionamiento del sistema propuesto.

1. Se genera una base de datos en modo de operación normal
2. Se genera una base de datos en modo de falla
3. Se entrena a las Máquinas de Soporte Vectorial
4. Se evalúa su desempeño, si no es aceptable regrese al paso anterior, si es aceptable continúe
5. Se toman datos de prueba
6. Se prueban los datos con MSV, si no detecta alguna falla regrese al paso anterior, de lo contrario continue
7. Diagnóstico, muestra la variable en falla y su localización.

4. Caso de estudio

Seis variables son supervisadas, estas corresponden a seis sensores encargados de monitorear lo siguiente:

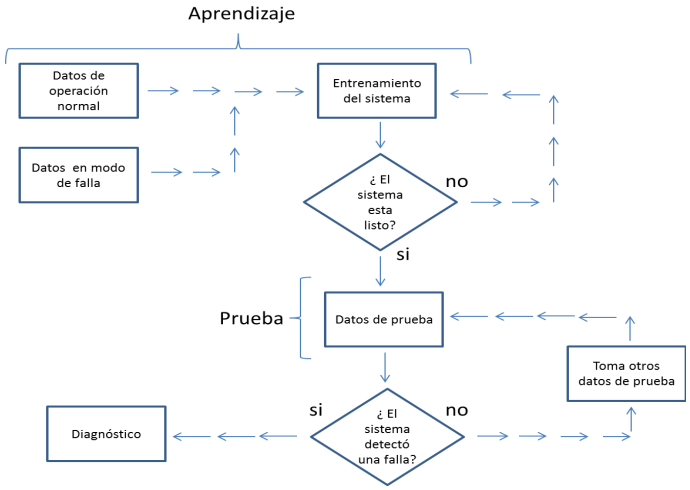

Fig. 3. Metodología basada en MSV.

1. Velocidad angular de la llanta frontal izquierda (FI)
2. Velocidad angular de la llanta frontal derecha (FD)
3. Velocidad angular de la llanta trasera izquierda (TI)
4. Velocidad angular de la llanta trasera derecha (TD)
5. Velocidad Longitudinal (VL)
6. Giro del vehículo sobre su propio eje (Yaw)

Las simulaciones se llevaron a cabo sobre la maniobra vehicular conocida como chicane. Chicane se refiere a un cambio en la trayectoria recta del vehículo, por ejemplo, cuando el automóvil realiza una maniobra de rebase. En el análisis realizado por [1] se encontró que para la variable yaw solo es necesario monitorear que en todo momento su valor sea diferente de cero, de lo contrario existirá una falla. Por lo tanto el análisis propuesto se llevará a cabo con las cinco variables restantes. Por lo descrito anterior se concluye que el análisis cuenta con seis estados posibles, operación normal y cinco estados de falla correspondientes a las cinco variables en análisis. La idea central del sistema propuesto es utilizar MSV como clasificador para poder detectar y localizar alguna posible falla, sin embargo como se mostró en la sección 2.1 las MSV estándar clasifica únicamente entre dos clases, por lo tanto se debe utilizar el enfoque de MSV multiclase (MSVMC) y así clasificar en seis distintas clases. Las cuales corresponden a cada uno de los estados mencionados anteriormente.

4.1. MSV multiclase (MSVMC)

Entre los métodos más aplicados para lograr una clasificación multiclase con MSV se encuentran los algoritmos conocidos como: uno contra todos (UCT)

el cual construye M MSV donde M es igual al número de clases a clasificar, después cada uno de estos SVM separa una clase del resto. Es decir el i-ésimo MSV es entrenado con todas las muestras de entrenamiento de la i-ésima clase con una etiqueta distinta a las otras. Otro método es el conocido como uno contra uno (UCU), el cual construye $M * (M-1)/2$ MSV. Estos combinan su función de clasificación para determinar la clase a la que pertenece la muestra de prueba, esto mediante la acumulación de predicciones de clasificación (votos), la predicción con más votos corresponderá a la clasificación final [11,12]. La Figura (4) muestra gráficamente estos dos métodos, las áreas rojas corresponden a las regiones no separables de cada método. Esto no se detalla en este artículo.

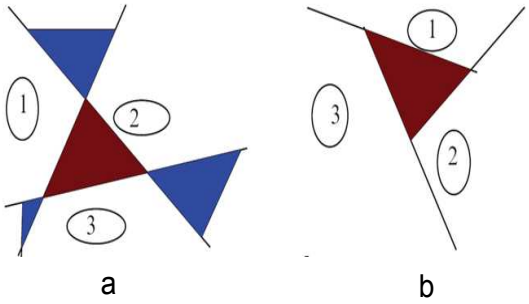

Fig. 4. a) UCT, b) UCU, adaptado de [12].

En esta propuesta se utilizará MSVMC usando el algoritmo UCU.

5. Análisis y resultados

Se procede a generar las bases de datos tanto en modo de operación normal como en modo de falla, la primera se compone de 4400 observaciones para las 5 variables, la segunda se construye a partir de datos para cada una de las fallas. Se tomaron 1600 observaciones para cada variable en modo de falla, generando una matriz de tamaño 12400×5. Para entrenamiento y validación del sistema se utilizó validación cruzada.

Como datos de entrada se utilizan los datos de entrenamiento antes descritos, además de un vector de etiquetas de tamaño 12400×1, el cual contiene seis etiquetas correspondientes a los seis estados posibles. El tiempo aproximado de cómputo para el entrenamiento fue de 6.3 segundos, mientras el desempeño del clasificador fue del 99.95 %. El modelo para la clasificación utilizó un kernel radial y encontró un total de 1612 vectores de soporte. La Tabla 1 muestra la matriz de confusión resultante de la validación del sistema.

Se puede observar que el algoritmo clasifica un dato erróneamente, de 333 datos correspondientes a la clase 4 clasifica un dato de ellos en la clase 0, aun

Tabla 1. Matriz de confusión del clasificador MSV.

	0	1	2	3	4	5
0	899	0	0	0	0	0
1	0	313	0	0	0	0
2	0		324	0	0	0
3	0	0	0	331	0	0
4	1	0	0	0	322	0
5	0	0	0	0	0	290

así su desempeño es bastante bueno por lo tanto se prosigue con la siguiente etapa del sistema. A continuación se obtienen observaciones para su monitoreo, del total de estas se toman ventanas de 100 muestras para su análisis. Dicho tamaño de muestra se utilizó al encontrar que es un valor estándar empleado y recomendado en la revisión del estado del arte [1].

5.1. Presencia de falla

Se considera que existe falla en un sensor cuando en él se presenten cambios en su valor nominal fuera de un rango de $\pm 2\,\%$. La Figura 5 muestra una falla presente en el sensor encargado de monitorear la velocidad angular de la llanta delantera izquierda. Esta falla representó un cambio de menos 5 % en la medición con respecto a su valor nominal y se presentó durante 12 unidades de tiempo (segundos). Al monitorear estos datos con el sistema propuesto, éste detecta la falla en la variable 1 (FI) a partir de la observación 21 y durante 12 unidades de tiempo. Como se mencionó anteriormente el análisis se lleva a cabo en ventanas de 100 datos. En el análisis anterior la ventana se tomó a partir de la muestra 160 hasta la 259, lo cual corresponde con la localización mostrada por el sistema.

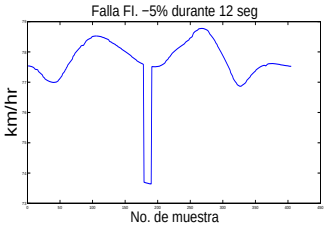

Fig. 5. Falla presente en sensor de llanta delantera izquierda.

A continuación se muestran los resultados obtenidos por el sistema propuesto en este artículo en comparación con el sistema propuesto por [1]. Tal comparación se obtuvo al realizar doscientas simulaciones de fallas simples (presencia de falla en una sola variable a la vez) en distintas localizaciones y a diferentes magnitudes

de desviación con respecto a los valores nominales de las variables monitoreadas. En las siguientes tablas se puede observar la comparación de la nueva propuesta contra la metodología empleada en [1]. La Tabla 2 muestra los porcentajes de detección, la Tabla 3 los porcentajes para la identificación, la Tabla 4 lo hace para la localización y la Tabla 5 muestra las comparaciones cualitativas de ambos métodos.

Tabla 2. Exactitud de algoritmos en detección de fallas.

Desviación	Muestras en falla	Detección con MSV	Detección con [1]
5 %	15	100 %	100 %
5 %	10	100 %	96.66 %
5 %	5	100 %	66.66 %
3 %	15	100 %	76.66 %
3 %	10	100 %	63.33 %
3 %	5	100 %	36.66 %

Tabla 3. Exactitud de algoritmos en identificación de fallas.

Desviación	Muestras en falla	Identificación con MSV	Identificación con [1]
5 %	15	100 %	93.33 %
5 %	10	99.6 %	86.66 %
5 %	5	100 %	51.66 %
3 %	15	99.4 %	65 %
3 %	10	100 %	55 %
3 %	5	100 %	21.66 %

Tabla 4. Exactitud de algoritmos en Localización de fallas.

Desviación	Muestras en falla	Localización con MSV	Localización con [1]
5 %	15	100 %	91.66 %
5 %	10	100 %	90 %
5 %	5	100 %	56.66 %
3 %	15	100 %	71.66 %
3 %	10	100 %	61.66 %
3 %	5	100 %	26.66 %

La gran eficiencia de clasificación de las máquinas de soporte vectorial así como su capacidad de trabajar con gran número de datos da como resultado un

Tabla 5. Características de ambos métodos.

Método propuesto (MSVMC)	Método [1]
• Necesita bases de datos en modo de operación normal y en modo de falla	• Solo necesita base de datos en modo de operación normal
• Requiere tiempo de entrenamiento	• Capaz de aprender apartir de los datos en operación normal
• Capaz de manejar ruido	• Capaz de manejar ruido
• Capaz de operar en linea	• Operación fuera de linea

desempeño superior del sistema propuesto en este artículo para la detección, clasificación y localización de fallas en comparación con el método propuesto por [1]. Debido a que este último utiliza solo datos de operación normal para su aprendizaje y su funcionamiento se basa en la variabilidad de los mismos. Dando como resultado un sistema el cual su exactitud depende del tamaño de muestra a analizar. Esto significa que el sistema es incapaz de trabajar en línea.

6. Conclusiones

En este artículo se presentó un sistema para la detección y diagnóstico de fallas para la dinámica lateral de un automóvil basado en Máquinas de Soporte Vectorial Multiclase. El desempeño de dicho sistema fue comparado con el método propuesto por [1], el cual combina Escalamiento Multidimensional, Rangos Percentiles y Análisis de Componentes Principales. Los resultados mostraron que el sistema propuesto tiene un mejor desempeño en tareas como detección, identificación y localización. Debido a la gran eficiencia de clasificación de las máquinas de soporte vectorial así como su capacidad de trabajar con gran número de datos. Además se observó que la exactitud del método basado en MSV no depende de la cantidad de muestras a analizar, por lo que al realizar pruebas con solo una observación los resultados fueron prácticamente iguales a los realizados con 15, 10 y 5 observaciones. Por lo tanto al respecto se concluye que el sistema propuesto es capaz de trabajar en línea, a diferencia del método [1]. Las principales desventajas del método con MSV frente al método mencionado, radican en la necesidad de bases de datos en modo de falla y al tiempo computacional generado por entrenamiento, siendo esta última menos importante ya que el tiempo de entrenamiento fueron aproximadamente seis segundos, un tiempo relativamente corto. De acuerdo a los resultados presentados en este artículo se concluye que las Máquinas de Soporte Vectorial son una técnica robusta para tareas de clasificación, y presenta excelentes resultados en el campo del diagnóstico y detección de fallas.

Referencias

1. Juan Pablo Nieto González, Luis Eduardo Garza Castañon, Abdelhamid Rabhi, Ahmed El Hajjaji: Fault Detection and Diagnosis of a Vehicle Combining Multidi-

mensional Scaling, Percentiles Range and PCA. In: 9th international conference on Sciences and Techniques of Automatic control & computer engineering, Túnez, pp. 1–11 (2008)
2. Baoling Liu, Dibo Hou, Pingjie Huang, Banteng Liu, Huayi Tang, Wubo Zhang, Peihua Chen, Guangxin Zhang. An improved PSO-SVM model for online recognition defects in eddy current testing. Nondestructive Testing and Evaluation, 28(4), 367–385 (2013)
3. Esraa Elhariri, Nashwa El-Bendary, Mohamed Mostafa M. Fouad, Jan Platos, Aboul Ella Hassanien, Ahmed M.M. Hussein: Multi-class SVM Based Classification Approach for Tomato Ripeness. Innovations in Bio-inspired Computing and Applications, Advances in Intelligent Systems and Computing, 175–186 (2014)
4. Amaury B. Andre, Eduardo Beltrame, Jacques Wainer: A combination of support vector machine and K-nearest neighbors for machine fault detection. Applied Artificial Intelligence: An International Journal. Vol. 27, 36–49 (2013)
5. N.M. Khan, R.Ksantini, I.S.Ahmad, B.Boufama: A novel SVM + NDA model for classification with an application to face recognition. Pattern Recognition, Vol. 45, 66–79 (2011)
6. Elkin Eduardo García Díaz: Boosting support vector machines. Tesis de maestria, Bogotá, Colombia, Universidad de los Andes, facultad de ingeniería (2005)
7. Gong Yanjie, Gao Xuejin, Wang Pu, Qi Yongsheng: Online Modeling Method Based on Dynamic Time Warping and Least Squares Support Vector Machine for Fermentation Process. In: 8th World Congress on Intelligent Control and Automation. inan, China, pp. 481–485 (2010)
8. Nicholas I. Sapankevich, Ravi Sankar: Time series prediction using Support Vector Machines. IEEE computational intelligence magazine, 24–37 (2009)
9. Hsu Chun-Chin, Chen Mu-Chen, Chen Long-Sheng: Intelligent ICA-SVM fault detector for non-Gaussian multivariate process monitoring. Expert Systems with Applications, Vol. 37, 3264–3273 (2010)
10. Zhiqiang Ge, Zhihuan Song: Multivariate Statistical Process Control. Springer-Verlag, London (2013)
11. Hassen Keskes, Ahmed Brahama, Zied Lachiri: Broken rotor bar diagnosis in induction machines through stationary wavelet packet transform and multiclass wavelet SVM. Electric Power Systems Research, Vol. 97, 151–157 (2013)
12. Oscar Castillo, Li Xu, Sio-Long Ao: Trends in Intelligent Systems and Computer Engineering. Springer Science + Business Media, New York, USA (2008)