

Descripción de metodologías de Machine Learning (ML) para la identificación de actividades a través de reconocimiento de patrones.

Description of Machine Learning (ML) methodologies for the identification of activities through pattern recognition.

Carlos Enrique Oballe Neyra¹, Xiomara de los Milagros Masias Rugel², Cristhian Nicolás Aldana Yarleque³
Instituto de Investigación en Economía y Eficiencia Productiva
Universidad Nacional de Frontera, Sullana, Piura, Perú.

RESUMEN

La aplicación de modelos de Machine Learning (ML) es cada vez más frecuente para la implementación, automatización y sistematización de procesos. No obstante, los modelos y técnicas disponibles en la literatura y desarrollo actual están diseñados para un mejor rendimiento, ya sea para potenciar la evaluación y clasificación de datos etiquetados, o la probabilidad de clasificar clústeres correctamente. El primero induce a mejorar la precisión en la evaluación de clasificación, ya que no se preocupa por el etiquetado, mientras que el segundo intenta mejorar la clasificación en datos que no están etiquetados. Debido a las ventajas y desventajas según la eficiencia de estos enfoques al uso extenso de datos, se usan modelos híbridos para maximizar la precisión en la clasificación, y en particular, el presente estudio realizó un análisis de cuatro enfoques ML de aprendizaje supervisado, implementado por la aplicación de algoritmos para seguir de cerca los procesos y comprender cada problema. Los resultados arrojaron una precisión variable entre 30% a 50% y una menor de modelos no supervisados.

Palabras claves: ML, aprendizaje semi-supervisado, clasificación, etiquetado.

ABSTRACT

The application of Machine Learning (ML) models is becoming more and more frequent for the implementation, automation and systematization of processes. However, the models and techniques available in the literature and current development are designed for better performance, either to enhance the evaluation and classification of labelled data, or the probability of classifying clusters correctly. The former induces improved accuracy in classification evaluation, as it does not care about labelling, while the latter attempts to improve classification in data that is not labelled. Due to the advantages and disadvantages according to the efficiency of these approaches to the extensive use of data, hybrid models are used to maximize accuracy in classification, and in particular, the present study conducted an analysis of four ML approaches of supervised learning, implemented by the application of algorithms to closely follow the processes and understand each problem. The results showed a variable accuracy between 30% and 50% and a lower accuracy of unsupervised models.

Keywords: ML, semi-supervised learning, classification, labelling.

¹ Instituto de Investigación en Economía y Eficiencia Productiva, Universidad Nacional de Frontera, Email: 2014101053@unf.edu.pe, <https://orcid.org/0009-0008-9600-0967>, Sullana, Piura, Perú.

² Instituto de Investigación en Economía y Eficiencia Productiva, Universidad Nacional de Frontera, 2019101034@unf.edu.pe, <https://orcid.org/0000-0001-5254-1165>, Sullana, Piura, Perú.

³ Instituto de Investigación en Economía y Eficiencia Productiva, Universidad Nacional de Frontera, caldana@unf.edu.pe, <https://orcid.org/0000-0002-6890-5370>, Sullana, Piura, Perú.

1. INTRODUCCIÓN

En la actualidad, el avance de la automatización de los procesos se ha generalizado en varias áreas de trabajo, desde mejorar la condición humana a través de soportes de dispositivos robóticos, actividades automáticas para procedimientos quirúrgicos o de suministro de medicina, hasta la implementación de sensores para rastrear patrones para mejorar la ergonomía en la actividad de los procesos de las fábricas.

Es así que, dentro de la investigación del reconocimiento de la actividad humana, este se ha tratado como un problema de reconocimiento de patrones el cual a avanzado por el continuo uso de algoritmos aplicado al aprendizaje automático, (Lara y Labrador, 2012). No obstante, el uso del método de aprendizaje como Decision Trees, Support Vector Machines (SVM), Hidden Markov Models entre otros, pueden ser satisfactorias en ciertos escenarios mientras que en otros traen consigo una serie de desventajas (Wang et al., 2019), ya que, para realizar una extracción de características adecuadas en entornos complejos y tareas generales, la dificultad de construir un sistema de reconocimiento de actividades aumenta (Bengio, 2013). Esto implica que solo se puedan aprender características muy superficiales (Yang et al., 2015) es decir actividades de bajo nivel como correr o caminar (Yang, Q., 2009), asimismo, el reconocimiento de patrones suele necesitar una gran cantidad de datos correctamente etiquetados para un buen entrenamiento del modelo, sin embargo, la mayoría de los datos de actividades en el contexto real no están etiquetados, lo que provoca disminución en el rendimiento de los modelos (Bengio, 2013) y por ende obligan a usar modelos de aprendizaje no supervisado los cuales presentan menor precisión en clasificación para una gran cantidad de datos.

En este estudio, se explicará la forma del preprocesamiento y procesamiento de los datos obtenidos, con el objetivo de realizar un análisis sobre algunos métodos muy usados para predicciones sobre un conjunto de datos de entrada que identifique correctamente la clase de la actividad real, pero sobre todo, crear las condiciones para que estudios con problemas en el procesamiento y predicción de modelos usando Machine Learning para el reconocimiento de patrones, tengan un nuevo contexto de ML aplicado que pueda facilitar la comprensión del uso de un método de aprendizaje semi-supervisado novedoso, a través de la explicación de dicho método de aprendizaje profundo el cual fue llevado en paralelo con la implementación del modelo por medio de cuatro enfoques.

Para la aplicación del preprocesamiento y procesamiento de data, se adaptó el algoritmo implementado para el reconocimiento de la actividad humana del trabajo fin de Máster, Aplicación del aprendizaje profundo al reconocimiento de la actividad humana (Piqueras, 2020). Donde se utilizó un subconjunto de datos medidos por sensores (acelerómetro, giroscopio, magnetómetro, orientación por cuaterniones), con datos provistos y facilitados para fines académicos y científicos, explicados técnica y detalladamente en el Dataset Manual Realistic sensor displacement benchmark dataset (Baños et al, 2012).

2.METODOLOGIA

2.1. Enfoques

El análisis se aplicó a 4 enfoques de Machine Learning dentro del aprendizaje supervisado: Fully Connected Deep Neural Network (FCDNN), Long Short – Term Memory, Convolutional Neural Network + Long short – Term Memory y Convolutional Long Short – Term Memory. Los enfoques mencionados son comúnmente utilizados para aprender patrones complejos. Sus diseños incluyen una capa de entrenamiento conformada por un conjunto de neuronas que se ocupan de tomar individualmente particularidades diferentes, las cuales son aprendidas a través de la identificación de patrones en las siguientes capas, configuradas con una función de activación determinada para el propósito en cuestión, responsable de manejar la no linealidad y resolver problemas complejos a diferencia de otros modelos de clasificación.

Asimismo, dentro del aprendizaje no supervisado se analizaron algunos enfoques de Machine learning; estos incluyeron el método de los k vecinos más cercanos (K Nearest Neighbors, KNN), árbol de decisión (Decision Trees), clasificadores Bayesianos Ingenuos (Naive Bayes) y Random Forest.

Por último, dentro del aprendizaje semi-supervisado se analizaron métodos como Support Vector Machines (SVM), Análisis de Componentes Principales (PCA) y un novedoso desarrollo dentro de la categoría del semi-supervisado con características de aprendizaje reforzado conocido como Hierarchical Extreme Learning Machine (HELM) desarrollado por Yao et al., (2018).

2.2. Preprocesamiento

El preprocesamiento de los datos recopilados en bruto es la parte más importante del proceso completo de clasificación por medio de aprendizaje de patrones con inteligencia artificial, esto se debe a que los datos que el sensor registra incluyen ruidos por la sensibilidad del propio sensor, errores mecánicos y eléctricos, fallas en la batería, entre otros. Los algoritmos y resultados analizados, fueron previamente computarizados en tiempo real para aumentar un mejor rendimiento al relacionar las actividades y los datos obtenidos a través del pre-procesamiento de la data almacenada y etiquetamiento, usando el software CRN Toolbox, (Baños et al., 2012).

A pesar de la existencia de limitaciones en el acceso del método que usa CRN Toolbox, se describe este concepto como un repositorio de software de componentes parametrizables incluyendo dispositivos de lectores y escritores I/O, filtros, componentes para dividir, y mezclar data, así como sincronización de flujo de data, editor para especificación de data y configuración de componentes y un espacio para temporizar la ejecución de configuración de componentes de software, entre otras herramientas externas que se comunican con el temporizador que sirve para que los datos recopilados sean pre-procesados en tiempo real a la clasificación para su procesamiento, (Bannach et al., 2008).

Mencionado preliminarmente, los datos se preprocesaron para eliminar alguna irregularidad en los datos y tener un mejor manejo al ser escalados y normalizados, existen varias técnicas para escalar los datos, estos son transformación lineal entre 0 y 1 desarrollado por Lapedes et al. (1988), Transformación lineal entre números límites desarrollado por Srinivasan et al., (1994) y la normalización simple desarrollado por Lachtermacher et al., (1995). La elección del tipo de transformación dependerá de la volatilidad de los datos recopilados, pues si estos presentan altas variaciones, sería recomendable usar la transformación entre 0 y 1, mientras que para datos

menos volátiles sería entre números límites o el simple para evitar que se vuelva muy lineal y se pierdan patrones muy importantes. Dado que los datos serán procesados en los modelos mencionados, estos deben estar organizados para que encajen a los requerimientos de los modelos, por eso, estos son concatenados por filas, divididos como entrenamiento y prueba y seccionados por ventanas deslizantes que permita mostrar un patrón en conjunto con las etiquetas definidas y relacionadas previamente con los datos de entrada. En este artículo se implementó un conjunto de funciones que fueron creadas para navegar entre en el escritorio en cuestión y se pueda manejar las hojas de cálculo (EXCEL) como un todo, para dar como resultado cada requerimiento de los modelos. Este implemento se desarrolló como complemento con el fin de comprobar algunos métodos usados y encaminar la comprensión a detalle de los métodos empleados.

2.3. Procesamiento y modelado

En el procesamiento se definieron el número de datos tomados por cada ventana deslizante, (72 como ejemplo a diferencia de 200 tomado por Piqueras (2020)), así como los hiperparámetros los cuales son ajustados y pueden ser modificados para la precisión de los modelos. Asimismo, los modelos son definidos y configurados con parámetros necesarios para el aprendizaje, entre ellos: “filters”, “kernel_size”, “activation”, “Flatten”, “Dense” etc. Después los modelos son configurados al tipo de entrenamiento apropiado para el caso, se define el tipo de pérdida “loss”, la optimización “optimizer”, y métricas “metrics” para la precisión.

Una vez configurados, los modelos son ajustados y entrenados únicamente en base a los datos seleccionados para entrenamiento, el cual muestra gráficamente la evolución de las pérdidas por entrenar y la precisión. Por último, se crean las variables de predicción en cada modelo para que capture, a través de funciones predeterminadas del modelo, la etiqueta o valor en un nuevo subconjunto de datos como entrada las cuales vienen de la división inicial de los datos tomados como prueba, y el modelo es evaluado por su precisión a través de la etiqueta que predice y la que está registrada en la base de datos de prueba.

En el caso de los modelos no supervisados como los K vecinos más cercanos KNN o PCA, los datos son separados en “clusters” o grupos los cuales son enumerados o clasificados, dicha clasificación es definida por el autor, en este caso 33 actividades, dándole el número 33 a un objeto de clasificación de KNN y creando 33 dimensiones para PCA o SVM. En estos modelos también se aplican hiperparámetros de filtrados, normalización, pero la minimización de pérdidas “loss” es diferente, ya que utilizan la métrica de la distancia euclidiana, Manhattan y Minkowski. Dado que se busca predecir clasificaciones de inputs no etiquetados, los resultados, aplicando los modelos no supervisados, arrojaron predicciones inestables en el presente artículo, esto se debe a que el modelo solo se guía por los patrones para determinar el error de clasificación o agrupación de manera que el cluster seleccionado fue impreciso.

2.4. Modificaciones del algoritmo de aprendizaje supervisado

A diferencia del algoritmo de Piqueras (2020), varias modificaciones fueron hechas a lo largo del algoritmo. En primer lugar, se hizo una pequeña modificación en la primera función, definida para navegar por el escritorio personal y no en el laboratorio virtual de Google Colab con acceso a Google Drive como en el algoritmo original.

Otra modificación surge de la idea de predecir o identificar datos que serán recolectados en el futuro. Debido que el arreglo inicial del algoritmo original se hizo para predecir un subconjunto

aleatorio de datos tomados como prueba, la predicción de las etiquetas se hace solo a datos que están etiquetados (aprendizaje supervisados). En el presente algoritmo la modificación implica la creación de un archivo Excel adicional donde se registre nuevos valores que los sensores registren en el futuro, la entrada de datos en el algoritmo también se alimentará de este nuevo archivo el cual será preprocesado, no obstante, los datos que se tomarán como prueba no será de forma aleatoria sino, serán los datos de este nuevo archivo y el cálculo para el total de datos de prueba está definida por el cociente del total de los datos futuros sobre el total de los datos recolectados.

A pesar que los modelos mencionados (mayormente en aprendizaje supervisado) están enteramente diseñados para evaluar su precisión de valores etiquetados o clasificados completamente registrados, para este algoritmo se pretende forzar la predicción de valores que no están clasificados y resolver el problema de la identificación remota de movimientos por sensorización. Para hacerlo, se registra intencionalmente etiquetas erróneas a los datos de prueba ya que no hay etiquetas, para que el algoritmo arroje en código binario su predicción en base a los datos entrenados. La modificación se hace a través de un bucle que extraiga las predicciones de etiquetas los cuales pueden ser verdaderos positivos, verdaderos negativos, etc.

Por lo general existen desventajas o riesgos en disminuir la precisión del modelo al no estar adaptado para este planteamiento, no obstante, obtendremos un nivel de confianza (en porcentaje) para cada valor o etiqueta del primer grupo de etiquetas arrojado por los modelos como predicción. Luego cada etiqueta o valor de clasificación obtenido reemplazarán las etiquetas erróneas registradas intencionalmente y se evaluará nuevamente para observar la precisión con estas nuevas etiquetas a través de otro bucle y el modelo predecirá un segundo grupo de etiquetas con sus respectivos niveles de confianza, y este proceso se iterará n veces hasta que la precisión de los modelos y nivel de confianza de las etiquetas converjan a niveles altos.

Se ha adoptado este planteamiento con la intención de mejorarlo a través de la inclusión y/o combinación de modelos no supervisados como K-Nearest Neighbor Classifier o K-means, a pesar de demostrar ser menos precisos por la no linealidad.

Como resultado de la utilización del algoritmo de aprendizaje supervisado por los 4 enfoques modificado, se halló que aún existe una baja precisión entre 30% y 50%, lo cual indica una mejora que cualquier técnica de aprendizaje no supervisado, pero baja. Una de las causas es porque se controlaron los datos que fueron divididos “Split” para entrenamiento y predicción, los cuales no se mezclaron y dificultó la mejora del rendimiento en el aprendizaje por patrones, trayendo consigo disminuciones considerables en la precisión de las nuevas predicciones. Otra causa importante fue que aun sin la modificación del “Split”, tras la mejora considerable de la predicción al reducir el número de ventanas deslizantes a 72 para mejorar el aprendizaje patrón-tiempo, existían pocas actividades que no tenían una buena predicción por defecto, lo cual lleva al presente artículo a informar y dar continuidad con las demostraciones para mejorar ya que a que existe evidencia de mejoras en la precisión.

A pesar de las limitaciones, se espera que crezca el potencial de generar mayor precisión que incluso los modelos no supervisados y los semi-supervisados clásicos, ya que estos últimos, aunque sean apropiados para predecir valores fuera del rango supervisado, sus precisiones se ven afectada por el nivel de dimensionamiento o número grande de clasificaciones, donde los modelos supervisados tienen una mayor ventaja.

2.5. Descripción matemática de los modelos de aprendizaje supervisado

La mayoría de los modelos de aprendizaje supervisado orientan su mecanismo de aprendizaje de patrones a través de las siguientes expresiones. Primero se obtienen las predicciones continuas convertidas a predicciones binarias tal como se observa en la siguiente relación:

$$\begin{aligned} f(x_i) &= \hat{y}_i = M(x_i), \\ i &= 1, 2, 3, \dots N. \end{aligned} \quad (1)$$

Donde M es la matriz de las predicciones realizadas por los modelos de clasificación, los cuales usan una función de pérdida de entropía cruzada categórica “Categorical Crossentropy” que evalúa la discrepancia entre las etiquetas reales y las predicciones de un modelo de clasificación múltiple cuya forma se presenta por medio de la siguiente expresión:

$$CC(y_{true}, y_{pred}) = - \sum_{j=1}^N y_{true,j} \cdot \log(y_{pred,j}) \quad (2)$$

N : es el número de clases

$y_{true,j}$: es la probabilidad real de la clase (j)

$y_{pred,j}$: es la probabilidad predicha de la clase (j)

Para la implementación del algoritmo del presente artículo, se argumentó que la validación se hace entre el valor predicho y el valor predicho anterior para obtener una predicción fuera del rango supervisado, de esta manera la ecuación queda expresada como:

$$CC(y_{i-1}, y_i) = - \sum_{j=1}^N y_{i-1,j} \cdot \log(y_{i,j}) \quad (3)$$

Con $f(x_i) = y_i$, donde y_{i-1} no necesariamente es la etiqueta real sino la predicción anticipada que es evaluado con una segunda predicción y_i , de tal forma que entre cada evaluación de predicciones se tiene un riesgo no controlado, el cual puede disminuir a través de la predicción promedio de los 4 modelos de clasificación y su nivel debe ser alto ante aumentos de la desviación de f debido a la existencia de una mayor incertidumbre en los modelos.

Después de ejecutar el modelo y obtener el último resultado de los cuatro enfoques implementados en este artículo al igual que el estudio de Piqueras (2020), en este artículo se extendió el proceso creando umbrales de evaluación para que el modelo genere predicciones incluyendo etiquetas erradas a los nuevos inputs. Estos llevaron a crear matrices de predicción binaria, con el fin de tener una predicción autónoma del modelo a través de un bucle donde el modelo prueba continuamente cada etiqueta con mayor probabilidad de ser la correcta. A continuación, se muestran algunos objetos creados y aplicados en el algoritmo después de la ejecución de los modelos y la obtención de los resultados:

- Umbral (Threshold): Primero, se define un umbral (en este caso, 0.5). Este umbral se utiliza para convertir las predicciones continuas en predicciones binarias. Si el valor de la predicción es mayor que el umbral, se clasifica como 1; de lo contrario, se clasifica como 0.

- Predicciones Binarias: Luego, se aplica el umbral a las predicciones continuas (`y_test_predicted_conv1stm`). Si una predicción es mayor que 0.5, se asigna 1; de lo contrario, se asigna 0. Esto crea un conjunto de predicciones binarias.
- Impresión de Predicciones Binarias: a través de este objeto se imprime las predicciones binarias utilizando `print(predicciones_binarias)`.
- En esta parte del algoritmo, se toman las predicciones continuas de un modelo (como las salidas de una red neuronal) y las convierte en predicciones binarias utilizando un umbral específico. Esto es útil para problemas de clasificación en los que se desea establecer un límite para la decisión de clasificación.
- `rounded_predictions4`: a través de este objeto, se redondean las predicciones continuas (`y_test_predicted_conv1stm`) a valores enteros. Esto significa que si una predicción es mayor o igual a 0.5, se redondea a 1; de lo contrario, se redondea a 0. Estas predicciones redondeadas se almacenan en la variable `rounded_predictions4`.
- `precision4`: Luego, se calcula la precisión (accuracy) comparando las predicciones redondeadas (`rounded_predictions4`) con los valores reales del conjunto de prueba (`y_test_conv1stm`). La precisión mide qué tan bien el modelo clasifica correctamente las instancias. Es la proporción de predicciones correctas en comparación con el total de predicciones.

En resumen, este fragmento de código evalúa la precisión del modelo al comparar sus predicciones redondeadas con los valores reales.

2.6. Modelo de aprendizaje semi-supervisado (helm)

El modelo Hierarchical Extreme Learning Machine (HELM) desarrollado por Yao et al. (2018), Es un modelo de aprendizaje profundo de patrones utilizando una estructura de red profunda de autocodificadores implementado para extracción de características no supervisado, y luego generar un aprendizaje extremo de máquina con un método de regularización introducido a través de un modelo de entrenamiento semi-supervisado, el cual permite no solo extraer de manera profunda información que la data contiene, sino también aprender más de la data extra sin etiquetar.

La representación matemática del modelo se describe por las siguientes ecuaciones:

$$H_i = g(H_{i-1} \circ W_i) \quad (4)$$

Donde

H_i : es el output de las capas escondidas i

H_{i-1} : es le output de la capa escondida anterior ($i-1$).

W_i : son los pesos de las capas que están entre $i-1$ e i (la actual y la anterior)

Asimismo, se muestra la función de error a minimizar la cual se expresa como:

$$\hat{\phi} = \frac{1}{2} \sum_{i,j} u_{i,j} ||\hat{y}_i - \hat{y}_j||^2 \quad (5)$$

De acuerdo a Yao, la introducción de un mecanismo de aprendizaje no supervisado adicional para determinar la distribución de las muestras, mejoraría el rendimiento de predicción del modelo. Por medio de la regularización múltiple, que se deriva del aprendizaje múltiple se obtienen la distribución de los datos con todas las muestras disponibles. Y en la construcción del modelo se asume que, similares muestras tanto en muestras etiquetadas:

$$S_l = \{x_i, y_i\}_{i=1}^l \tag{6}$$

Como en muestras no etiquetadas:

$$S_u = \{x_i\}_{i=1}^u \tag{7}$$

Están lo suficientemente cerca, donde los datos más similares se incorporan en una misma distribución, la cual se pretende minimizar a través de la función de error.

\hat{y}_i, \hat{y}_j : son los outputs predichos respecto a x_i, x_j

$u_{i,j}$: es la similitud entre x_i y x_j

Estas ecuaciones nos dicen que, para cualquier input sin etiquetar, existe al menos un input con etiqueta que presenta características similares, de tal forma que pueden adoptar características similares y por lo tanto patrones comunes en la muestra total de datos después de la inclusión. Ante la incorporación de los inputs no etiquetados, se puede aprovechar el conocimiento aprendido de los datos etiquetados para usarlo en etiquetar los datos no etiquetados, además, los datos no etiquetados serán agrupados con datos etiquetados según sus similitudes de tal manera que minimice el error (mayor similitud entre el grupo de datos x_i y x_j) y se pueda identificar la etiqueta y correcta.

Para tal fin, según Yao los pasos a seguir son los siguientes:

Se seleccionan las variables de acuerdo al análisis teórico y experiencia del operador. Luego se determinan los grupos de datos a entrenar y a validar. Después, se normaliza la data, parámetros y coeficientes, luego se determina el número de capas y neuronas en cada capa para el modelo HELM. A continuación, se inicializa aleatoriamente los pesos de los inputs y neuronas escondidas que servirán de soporte para ponderar el entrenamiento automático multicapa que es el siguiente paso, basado en Extreme Learning Machine (ELM) con todas las muestras etiquetadas y sin etiquetar, del cual se obtendrá como siguiente paso las matrices de pesos para cada capa, los cuales serán necesarias para el paso del entrenamiento semi-supervisado (ELM) para la última capa y se obtenga el peso de salida, y en paralelo se construirá el modelo HELM tomando las ecuaciones como referencia para la predicción y evaluación del rendimiento y finalmente predecir la variable de etiqueta.

3.RESULTADOS

Entre los resultados hallados por Yao et al., (2018) se encontró que el modelo HELM registró el menor valor de error cuadrático RMSE para los datos de entrenamiento con 0.0029 de error y 0.0033 de error en datos de prueba en comparación con 4 modelos más, cuyos valores más altos fueron 0.0052 y 0.0057 de error en entrenamiento y prueba respectivamente.

Cabe resaltar que la inclusión de etiquetas erróneas para los nuevos inputs no etiquetados significó una reducción en el rendimiento en los parámetros del modelo para minimizar el error, lo que implicó un aumento en la inestabilidad de precisión en el testeo de los datos debido a la forzada utilización del proceso de prueba del modelo para predecir con los últimos inputs. Esto se tradujo en que la precisión de las clasificaciones correctas estuviera por debajo del 50%.

4. CONCLUSIONES

Existen varias metodologías dentro del campo de Machine Learning que se usan de acuerdo al tipo de datos, a la cantidad y éstas siempre tienen ventajas y desventajas sobre las otras. En el caso de aprendizaje de patrones para clasificar actividades, no existe aún una técnica concreta de aprendizaje artificial que permita superar los problemas con la precisión debido a diversos factores que influyen en la toma inicial de data, la cual es preprocesada con ruido por fallas en los sensores, similitud de los datos de diferentes actividades, entre otros vistos a lo largo del presente artículo. Aunque se percibió un aumento de precisión en el uso de los modelos supervisados en un 30% a 50% de precisión obtenido, estos suelen ser muy bajos para establecerlos como una clasificación verdadera, aun así, existe esta pequeña mejoría aun sabiendo que se controlaron hiperparámetros como el “Split” y la inclusión de etiquetas erróneas los cuales reducen el rendimiento de la precisión por no mezclar la data en su totalidad para entrenamiento y predicción, como consecuencia, se permitió ampliar las posibilidades de mejora del desarrollo del modelo implementado en el artículo en un futuro cercano dependiendo de la continuidad de la mejora del modelo.

Por último, el modelo de aprendizaje semisupervisado de Yao et al., (2018) incrementó la posibilidad de mejorar el modelo por medio de la minimización de una función de error que intente igualar los inputs etiquetados con los inputs no etiquetados dado que deberían encontrarse cercanos e incluidos en la misma distribución, aspecto clave para determinar la orientación del input no etiquetado a una posible etiqueta correcta. Este modelo usa un aprendizaje semisupervisado para extraer información de los inputs no etiquetados a través de un aprendizaje extremo profundo el cual arrojó excelentes resultados en sus investigaciones.

REFERENCIAS BIBLIOGRÁFICAS

- Bannach et al., (2008). Rapid Prototyping of Activity Recognition Applications. *IEEE Pervasive Computing*, 1-19.
- Baños et al., (2012). A benchmark dataset to evaluate sensor displacement in activity recognition. *. In Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 1026-1035.
- Bengio. (2013). Deep learning of representations: looking forward. *International Conference on Statistical Language and Speech Processing*, 1-37.
- Lachtermacher et al., (1995). Backpropagation in time-series forecasting. *Journal of Forecasting*, 14, 381-393.
- Lapedes et al., (1988). How neural nets work In: Anderson (Ed), *Neural Information Processing Systems. American Institute of Physics*, 442-456.
- Lara, O., & Labrador, M. (2012). A survey on human activity recognition using wearable sensors. *IEEE communications surveys & tutorials*, 15(3), 1192-1209.
- Piqueras. (2020). Aplicación del aprendizaje profundo al reconocimiento de la actividad humana. *Máster Universitario en Inteligencia Artificial*. Valencia, España: Universidad Nacional de La Rioja (UNIR).
- Srinivasan et al., (1994). A neural network short-term load forecaster. *Electric Power System Research*, 28, 227-234.

- Wang et al., (2019). Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, 119, 3-11.
- Yang et al., (2015). Deep convolutional neural networks on multichannel time series for human activity recognition. *Twenty-Fourth International Joint Conference on Artificial Intelligence*.
- Yang, Q. (2009). Activity recognition: Linking low-level sensors to high-level intelligence. *Twenty-First International Joint Conference on Artificial Intelligence*.
- Yao et al., (2018). Deep learning of semisupervised process data with hierarchical extreme learning machine and soft sensor application. *IEEE Transactions on Industrial Electronics*, 65(2), 1491-1498.