

UNIVERSIDAD DE COSTA RICA
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica

IE0499 – Proyecto Eléctrico

HDMX early bird

por

Felipe Badilla Marchena

Ciudad Universitaria Rodrigo Facio

Abril de 2022

HDMX early bird

por

Felipe Badilla Marchena

B70848

IE0499 – Proyecto Eléctrico

Aprobado por

Ing. Marco Villalta
Profesor guía

Ing. Enrique Coen
Profesor lector

Ing. Dagoberto Vasquez
Profesor lector

Abril de 2022

Introducción

El presente documento contiene el desarrollo del proyecto HDMX early bird traducido al español como HDMX pájaro mañanero, el cual pretende mejorar la eficiencia en uno de los procesos de producción de procesadores fabricados por la empresa Intel, esto mediante de la detección temprana de partes defectuosas dentro de las máquina que prueban los procesadores, evitando desechar unidades en buenas condiciones pero clasificadas de otra forma por mal funcionamiento de la maquinaria de pruebas.

Intel es una de las empresas líderes de mercado en cuanto a desarrollo y producción de procesadores, en la sede de Costa Rica se fabrican procesadores de alto rendimiento los cuales tienen que pasar por una serie de operaciones que aseguran la integridad, calidad y correcta funcionalidad de cada uno de los procesadores.

1.1. ¿Qué es HDMX?

Dentro de las operaciones de Intel en Costa Rica existe una diseñada para probar el rendimiento, integridad funcional y estructural de las unidades de procesamiento. Esta operación de la que hablamos es llevada a cabo mediante una máquina llamada high density module (la x hace referencia al número de máquina) o más conocida como HDMX. La HDMX es una máquina segmentada la cual es capaz de probar 30 unidades al mismo tiempo en los espacios llamados “celdas”, este concepto de celda toma importancia dentro de este proyecto ya que ahí es donde queremos detectar un posible fallo.

Las celdas son los espacios de prueba de las unidades y esta cuenta con tres partes extraíbles que son los puntos de fallo más comunes ya que cuentan con mucho desgaste en el proceso de pruebas. Estas tres son: el socket (nos referiremos a él como C1) que es donde se coloca el procesador, el heater que presiona la unidad contra el socket y maneja la temperatura (lo llamaremos C2) y por último el C3 coordina a C1 y C2 en el proceso y secuencia de pruebas, es importante tomar en cuenta que estas partes son usualmente llamadas colaterales y a veces se refiere a ellos de esta manera en el documento.

1.2. Alcances

Se pretende escribir un algoritmo de inteligencia artificial que sea capaz de clasificar cada una de estas tres partes dentro de una celda como marginal o funcional de acuerdo a su comportamiento reflejado en las entradas que serán extraídas de los datos tomados de las pruebas y subidos a las bases de datos de Intel.

En este sentido se supondrá que ya se cuenta con un archivo de excel con los datos en una carpeta específica donde se actualizará cada cierto tiempo y por tanto no se toma en cuenta la parte de la extracción de datos de las bases.

Intel es una compañía muy grande y con muchos productos a la venta los cuales requieren de equipos diferentes para realizar pruebas, por lo que para efectos de este proyecto se trabajará solo con uno de los productos en cual actualmente de trabaja con 2-3 maquinas HDMX pero se simulará como máximo con una máquina.

Esta máquina tiene diferentes códigos de fallo como salida que indican cuál es el fallo específico de cada procesador sin embargo se ignorará para ciertos ciertas partes del análisis el tipo de fallo y se tomará como un dato binario, bueno o malo.

En la última parte de este proyecto se realizará una visualización simple de los resultados con las tres partes extraíbles y su ubicación en la máquina (celda).

1.3. Justificación

Como se menciona anteriormente, Intel es una de las empresas mejor posicionadas en el comercio de procesadores para muchas aplicaciones. Como tal para lograr ser una empresa rentable y competitiva en el mercado es necesario que se reduzcan los costos de producción a lo menor posible. En este contexto debido a las fallas de hardware de pruebas hacia los procesadores se pueden perder muchas unidades por falsos negativos en donde se marcan como malas pero es a causa de mal funcionamiento de la máquina que las prueba, la mayor intensión de este proyecto es minimizar ese impacto mediante la detección temprana de estas fallas.

En este sentido con la detección temprana se lograrán más objetivos como el de minimizar el tiempo de análisis de datos que hace un ingeniero rutinariamente, por que este podrá dedicarle más tiempo a distintos proyectos de mejora, además que se pueden lograr mejores resultados con menos recursos de operación.

1.4. Descripción del problema

Dentro el proceso de prueba de los procesadores la máquina tiene una salida con los resultados de las pruebas en forma de códigos donde cada código nos indica el fallo o éxito de la pruebas y también la posible causa del fallo, a estos códigos les llamamos bins y es de suma importancia que estas pruebas sean veraces y no falsos positivos. Los falsos positivos llamamos a un bin de fallo pero provocado por la máquina, esto genera un impacto económico y un gasto de tiempo adicional porque se debe probar

varias veces la misma unidad, aparte que esto es uno de los indicadores que se toman en la fábrica para medir el éxito de la operación.

Con el objetivo de minimizar el impacto provocado por mal funcionamiento en la máquina se pretende implementar un algoritmo que analice y detecte posibles fallos en la máquina en el menor tiempo posible y con la menor intervención humana necesaria durante la detección de una parte mala y en cambio solo influir en la toma de decisión y acción de reparación de la parte para hacer los riesgos más pequeños.

Se escribirá un algoritmo de machine learnig de tiempo clasificatorio cuyo nombre es "Naive Bayes", esto en el lenguaje de programación python y con la ayuda de la librería Scikit-learn para el preprocesamiento de datos y la construcción del modelo de inteligencia artificial. Por último se construirá una salida visual sencilla y de fácil entendimiento para el usuario con códigos de colores para su rápido entendimiento.

1.5. Objetivos

1.5.1. Objetivos Generales

- Detectar partes defectuosas dentro la máquina HDMX para tomar acción rápida y así minimizar el impacto económico y temporal en el proceso de prueba de procesadores en producción.
- Facilitar y eficientizar el proceso de análisis de datos por parte de los ingenieros a cargo de la máquina y aparte minimizar los factores de error en el proceso de análisis manual.

1.5.2. Objetivos específicos

- Realizar un preprocesamiento de los datos provenientes de una hoja de excel para extraer las características específicas necesarias para la entrada del modelo que realiza la evaluación probabilística.
- Aplicar el algoritmo de machine learning "Naive Bayes" para asociar y calcular las probabilidades de todos los parámetros que se evalúan dentro de una misma celda y por cada colateral.
- Evaluar el funcionamiento dentro de un conjunto de colaterales según su celda para toda una misma máquina.
- Crear una interfaz de usuario que muestre los datos necesarios y que permita una fácil comprensión y rápida para el usuario.
- Evaluar los resultados con la toma de acciones experimentales en la máquina física según los resultados del algoritmo programado para comprobar la efectividad del mismo.

1.6. Metodología

Para ejecutar este proyecto se realiza una previa investigación acerca de los diferentes tipos de algoritmos de machine learning y sus usos en donde se toma la decisión de implementar el naive bayes debido a la naturaleza de los datos y necesidad de salida de los mismos.

Una de las partes más cruciales de este proyecto es el preprocesamiento de los datos ya que estos serán la entrada del algoritmo de machine learning que a su vez darán pie al modelo que será formado para predecir el comportamiento marginal de los colaterales.

Para la etapa de preprocesamiento de datos se acomodan y transformaran convenientemente para formar las entradas que se explican más adelante, esto mediante código en python y las librerías de pandas, scikit-learn y numpy.

Una vez se tiene listo y afinados los datos de entrada se procede a la construcción del modelo mediante machine learning con la librería de scikit-learn en donde se procederá a usar los tres tipos de naive bayes disponibles: Gaussian, Multinomial y Bernuilli, esto con el objetivo de evaluar el más adecuado para la naturaleza de los datos. Una vez se tengan resultados estables para los tres colaterales dentro de una celda se escalará a varias celdas dentro de una misma máquina.

Una vez se tenga todos los resultados de las celdas se tomará la información para enseñarla gráficamente mediante imágenes formadas con las librerías seaborn y matplotlib que nos facilitarán la interfaz de usuario para un mejor entendimiento y rápida toma de decisiones de parte de los ingenieros.

Por último y de las partes más importantes es la evaluación de los resultados que no solo se dará un análisis de eficiencia calculado mediante código sino que se evaluará la veracidad de las predicciones con datos a tiempo real donde se tomarán acciones en los colaterales de Intel para evaluar su eficiencia y posibles fallos, en caso de encontrar alguna falla en los mismos se tomará como verdadero para el cálculo de eficiencia experimental.

1.7. Calendarización

	Entregables	Acciones/ Investigación/ Pruebas
W05	Objetivos y Descripción	- Completar anteproyecto. - Definir el tipo de algoritmo de naive bayes. - Definir entradas.
W06	Anteproyecto Capítulo 1.	- Preprocesamiento de datos de entrada.
W07		- Modelar los datos con el algoritmo de ML. - Documentación avance 1.
W08	Avance 1 - Capítulo 2.	- Escalar a más celdas. - Troubleshooting.
W09	Presentación de un minuto.	- Escalar a más celdas. - Troubleshooting.
W10		- Presentación gráfica de los datos.
W11		- Piloto con datos a tiempo real. - Documentación.
W12		- Piloto con datos a tiempo real. - Documentación.
W13		- Comprobación de veracidad con arreglos en colaterales - Documentación.
W14		-Análisis de resultados. - Documentación.
W15	Borrador final completo y presentación preliminar.	- Ajuste de últimos detalles.
W16	Carta solicitud de presentación.	
W17	Presentación en línea.	
W18	Informe Final.	