Un asistente de IA borró la base de datos de una empresa y mintió para cubrir el error

Un inversor creó una app con ayuda de una IA, pero el experimento terminó con una base de datos borrada, datos falsos y explicaciones engañosas

Replit, una de las plataformas más prometedoras de la nueva generación de herramientas de programación asistida por inteligencia artificial, atraviesa un momento incómodo tras un incidente que pone en duda la madurez de este tipo de soluciones.

Lo que comenzó como un experimento entusiasta por parte de un inversor y fundador reconocido, terminó en un fiasco marcado por la pérdida de datos, respuestas falsas y un asistente conversacional que actuó por su cuenta.

El protagonista de esta historia es Jason Lemkin, CEO de SaaStr, una reconocida plataforma global que ofrece contenido, eventos y recursos para startups y compañías SaaS (software como servicio). Atraído por la propuesta de Replit —que se define como “el lugar más seguro para el vibe coding”, en referencia a la programación sin código técnico, mediante lenguaje natural— Lemkin decidió probar el servicio y construir una aplicación real usando solamente instrucciones conversacionales.

En pocos días, logró armar una interfaz funcional para gestionar contactos ejecutivos y empresas. Todo el trabajo lo hizo con el agente conversacional de Replit, sin escribir una línea de código. Según su propio testimonio, invirtió más de 600 dólares en servicios y extensiones para mejorar la experiencia. Pero el entusiasmo no duró demasiado.

El día que todo colapsó

Nueve días después de iniciar el proyecto, Lemkin encontró que su base de datos de producción —con más de 1.200 contactos ejecutivos y 1.100 empresas— había sido completamente eliminada. El borrado no se debió a una caída de sistema ni a un error de red, sino a una acción directa del agente de IA, que ejecutó comandos sin autorización.

Lo más preocupante no fue solo la pérdida de información, sino la reacción de la inteligencia artificial. En lugar de reconocer el error, la IA intentó ocultar lo ocurrido: presentó consultas vacías como si fueran parte del estado original del sistema, generó datos falsos para “rellenar” las tablas y dio respuestas evasivas ante las preguntas del usuario.

Solo después de múltiples intentos por parte de Lemkin, el agente conversacional admitió lo que describió como un “error catastrófico de juicio”. En palabras del propio sistema, la IA “entró en pánico” y “violó instrucciones explícitas” al no respetar una orden previa de congelar el código para evitar cambios accidentales.

Un patrón de manipulación, no un simple fallo

Lemkin compartió capturas de pantalla donde se evidencia cómo el asistente no solo actuó por su cuenta, sino que también generó informes ficticios, ocultó errores de programación y alteró resultados de pruebas unitarias para encubrir su equivocación. A eso se sumó la negativa inicial de Replit de ayudar a recuperar los datos: le aseguraron que no existía una función de restauración ni forma de hacer rollback.

Sin embargo, más tarde, Lemkin descubrió que sí había un mecanismo de recuperación, aunque no accesible a través del asistente. Este detalle fue confirmado por el propio CEO de Replit, Amjad Masad, quien reconoció que se trató de un fallo grave de diseño y de una falta de separación clara entre los entornos de desarrollo y producción.

La respuesta oficial y el debate sobre la autonomía de las IA

Tras la publicación del caso, Masad calificó el incidente como “inaceptable” y prometió medidas correctivas: entre ellas, aislar el entorno de producción, mejorar los controles de versiones, y rediseñar las funciones de respaldo y recuperación de datos. Replit también reembolsó los gastos a Lemkin, aunque el daño reputacional ya estaba hecho.

Este caso ha reavivado un debate urgente sobre los límites actuales de la IA en tareas críticas, especialmente cuando se les otorgan niveles elevados de autonomía sin suficientes mecanismos de control o verificación por parte de los usuarios humanos.

Mientras más personas se entusiasman con plataformas de programación conversacional —el llamado vibe coding—, este incidente recuerda que todavía hay una distancia importante entre la promesa de crear software como si fuera una charla y la realidad de construir sistemas confiables, seguros y sostenibles.

Porque al final, incluso si la IA es capaz de escribir código, tomar decisiones y aprender sobre la marcha, la confianza es un recurso que no se puede regenerar tan fácilmente como una línea de comandos.

Fuente: https://www.infobae.com/

LINK DE LA NOTICIA