ISSN: 2993-8643

Research Article

Engineering: Open Access

Implementing Centralized Error Handling for Software Systems through the Integration of Machine Learning Techniques

Thomson Alexander*

TMEIC

*Corresponding Author
Thomson Alexander,
TMEIC

Submitted: 2023, May 04; **Accepted**: 2023, Jun 08; **Published**: 2023 June 15

Citation: Alexander, T. (2023). Implementing Centralized Error Handling for Software Systems through the Integration of Machine Learning Techniques. *Eng OA, 1*(1), 01-08.

Abstract
Based on the increase of data volumes in the current world, modern software solutions' complexities and transaction volume make it imperative to establish more efficient and robust error-handling approaches. Traditional strategies have struggled to cope with the dynamics of voluminous transactions known to be decentralized and ad-hoc. This has led to operational disruptions and diverse software efficacy disruptions. The central prism of this paper points to how to leverage a proposed central error-handling system (CEHS). The proposition encapsulates how machine learning techniques can be leveraged to address these challenges. Discussing the limitations of current error-handling methods and highlighting benefits stemming from a centrist approach are embedded in this discourse. Consequently, this thread explores integrating ML algorithms as a prerequisite for identifying anomalies, triggering remedial actions, and predicting errors that could stem from the scenario. The remit of this discourse is to present a framework for CEHS implementation and discuss the potential impact regarding the reliability and performance of high transactional volumes through the prism of novel approaches in ML like data science skills and software engineering skills.

Keywords: Centralized error handling, machine learning, anomaly detection, proactive error mitigation, system reliability, operational efficiency, DevOps, CI/CD.

Introduction
Different software systems developed by various companies for processing high transactional volumes in e-commerce, finance, healthcare, or other sectors play crucial roles in catching errors. Their good work is very important because it can bring many problems if they fail. This could cost money, hurt their image, and stop services from working properly. In this situation, error handling plays a significant role. It helps make the system strong and smooth in processing transactions. Old ways of handling errors, which are usually disorganized and slow to respond, do not work well in today's busy systems [1]. These systems show built-in difficulties, with complicated relationships among parts and changing transaction movements. Also, there are problems in finding and fixing mistakes quickly because of the many deals that happen [2]. This causes late actions and more time when things aren't working correctly.

This paper suggests a new strategy - a central error handling system (CEHS) using machine learning (ML) methods to fix these problems. The CEHS wants to make mistake management better and use the future-predicting powers of machine learning. This will help quickly deal with errors in busy transaction systems like stores or banks.

Exploring Existing Research Regarding Error-Handling
Large computer systems used for many transactions can have weaknesses because things are connected and dependent on each other. A small problem with one part can spread, causing mistakes throughout the machine. The complex nature of these systems creates unexpected situations, making for unusual occurrences and surprising mistakes. Systems first made for smaller amounts might not work well when there's a lot of pressure. It is likely to culminate in errors when transactions are being done rapidly.

Errors could be embedded in large transactional financial systems especially when issues are encapsulated in the data. For instance, some elements could be missing in data or there could be misleading information like errors that occur later, or bad math. At the same time, data issues could occur simultaneously in the same system at various areas. In most cases, this scenario is likely to be linked to external sources significant in reducing the probability of mistakes.

Users of computer systems are usually likely to act as the cause of errors when numerous transactions are taking place. Mistakes done by users could include wrong input implicated in erroneous transactions. While this is true, in some cases, system-based issues could cause errors in light of passing huge financial transactions. When changing systems or setting them up, failure to take care of load capacity expected in a system carrying out large financial transactions can culminate in errors. For instance, errors stemming from financial calculations done by hand could impact the system negatively besides increasing the load regarding the number of financial transactions.

Moreover, network issues could also make large financial systems to face risks or inefficiencies. Network disruptions can perturb efficiencies when handling numerous transactions by causing loss of data or interruptions in connectedness for a seamless experience in processing data. When data gets lost, for instance, wrong results could arise. In other instances, hardware lapses could occur because of limited hardware resources that could cause computer systems to crash or lead to data loss. Power loss could also impact hardware systems negatively leading to errors in financial systems.

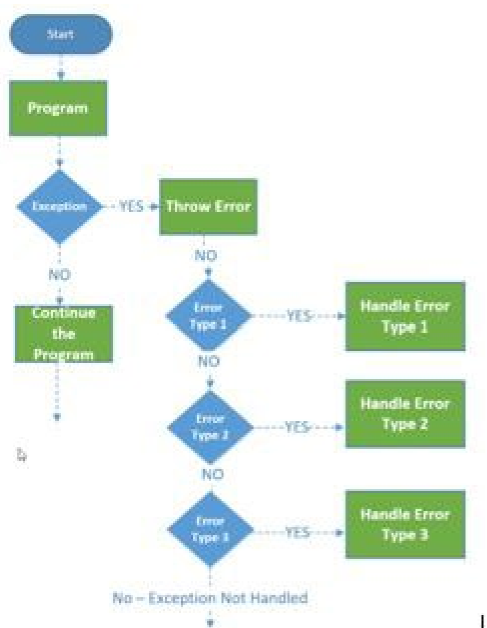

Figure 1

CEHS implementation success relies heavily on appropriate use of machine learning algorithms to find odd things, predict error, and work on finding error causation. Such a system is likely to bring to the front potential mistakes, or the desired results by deploying appropriate algorithms [3]. It can use data science-based algorithms like those used for regression analyses to find mistakes based on past information. At the same time, data clustering approaches are efficient at finding foreign aspects by capturing complicated data. IT teams should be bent on testing corner cases in computer programs deployed to determine whether they prevail in the worst case-scenario.

Traditional error handling is sequential in nature where we handle the error as and when it is required and logged to file or database. The exceptions that occurred in certain module will be thrown and handled based on the type of exceptions. If any specific error which is not handled separately will be logged as common exception. Refer Figure. 1

Making the main plan of CEHS involves knowing all parts and how they work together. The design should easily fit in and cause little trouble for what is already happening. Creating strong connections for sharing real-time data between the CEHS and other system parts is very important. This ensures information is always flowing, so errors can be fixed and solved quickly. Also, it's very important to consider how big the system can get and how easy it is to change [4]. This is because we need a structure that can deal with many complex things happening simultaneously over time.

Also, the main system needs ways to store data safely and quickly. This includes collecting mistakes reports and performance info in one place. This store of information is the base for training ML models. It helps them learn and change based on past examples.

Careful consideration is necessary while selecting appropriate machine learning algorithms and devising a centralized design for CEHS. Engaging in deep contemplation on the capabilities of algorithms and the functioning of system designs ensures that the CEHS (Computer Engineering and Hardware Systems) is robust in technology while also catering to the unique requirements of large-scale systems frequently operated throughout the day [5]. The success of the CEHS can be attributed to the collaborative efforts of intelligent machine learning algorithms and a well-designed central structure. These two components enhance error-handling abilities in intricate computer environments, increasing durability.

Challenges of Traditional Error-Handling

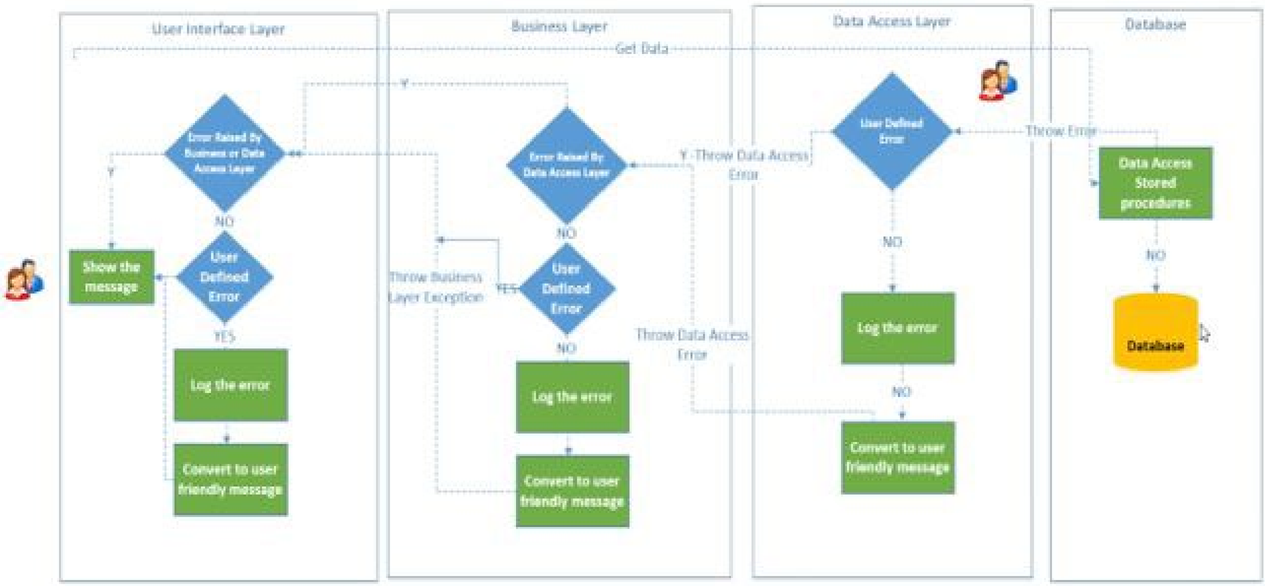

Figure-2

Old ways of fixing mistakes are hard because they don't work well in today's changing and complicated computer world. One big problem with these methods is that they can't grow much. As the systems get more complex and there are many transactions, manually writing down error cases becomes hard work. It's also easy to make mistakes in keeping up with rules for handling exceptions. This restriction makes it hard to run big systems smoothly and raises the chance of missing important errors. In the traditional approach the exceptions thrown from the database layer is thrown and send back to business layer and eventually to the UI layer. At UI layer separate conditions are handled and if the error is something appropriate to be displayed to user will be displayed otherwise will be logged. During the process error at various levels also can be converted to user friendly message as it propagates from one layer to other. Refer Figure.2

Another problem with traditional error handling is that it reacts when something goes wrong. These ways mainly deal with fixing mistakes once they have happened. They leave systems open during important times of finding and responding to errors. This quick reaction can cause longer shutdowns and weak system safety, as it doesn't immediately fix mistakes or stop them from hurting the computer's operation.

Another big problem comes from not taking action in the usual ways of dealing with mistakes. Not seeing errors before they happen makes it hard for us to stop them from getting worse. This means extra problems and less good results because we didn't catch the issues beforehand. If we can't guess and fix possible mistakes before they happen, our systems will be open to problems that could make users unhappy or slow down work.

Poor problem-solving makes fixing errors in the old way even harder. Finding and correcting errors in complex systems can take a lot of time and resources. Machine learning plays a crucial role in catching errors by deploying specific algorithms for this task. For instance, algorithmic formations like “try catch” are used by machine learning algorithms to detect and address errors in large data systems. This issue worsens when things are not working because good time is used to find and fix problems. Instead of making sure the system keeps running smoothly all the time.

These issues indicate the significance of fixing mistakes actively and completely primarily regarding numerous financial transactions. New ideologies can be invoked by engineers to resolve errors embedded in computer systems to increase efficiency. To this effect, a rigorous strategy is handy for careful error elimination whether the error is data-based or time-entrenched. This contributes to promoting the rigor of these financial systems so that there is more accuracy to yield increased customer satisfaction. When these steps are taken, they aid in mitigating potential financial drawbacks that could arise when operating a business.

Technical Aspects of Implementing a CEHS using ML

Effective CEHS practices are centric to technical aspects of its implementation. ML can be used to implement CEHS technical aspects by harnessing data science capabilities used in deep learning algorithms. The thoroughness comes into play when selecting machine learning algorithms or drawing from a specific central design. An in-depth exploration of the CEHS dynamics helps ensure effective working by fitting deftly in larger computer systems that handle vast financial transactions. In other words, the overarching technical aspects of using CEHS in machine learning are software engineering skills, and data science skills

Table 1:

Aspect	Traditional Approach	Machine Learning Benefit
Automated Issue Detection	Manual detection; time-consuming and prone to oversight	Automatic identification of patterns and anomalies for quicker and more accurate error detection
Real-time Error Monitoring	Periodic checks; may miss real-time issues	Continuous monitoring with real-time alerts for immediate intervention
Predictive Maintenance	Reactive measures after error identification; downtime	Predictive models anticipate errors based on historical data, enabling proactive maintenance
Adaptive Problem Resolution	Fixed rule-based systems may struggle with evolving or complex errors	ML algorithms adapt to changing conditions, learning from new data to improve error resolution over time
Reduced False Positives	Rule-based systems may generate false positives	ML models distinguish between normal variations and actual errors, reducing false positives
Anomaly Detection	Defining all possible error scenarios manually can be challenging	Anomaly detection models identify unexpected patterns or deviations, aiding in the discovery of novel errors
Customized Error Handling	Generic error-handling strategies may not be optimal for specific systems	ML models can be trained on system-specific data, allowing for customized error handling tailored to the system
Data-driven Insights	Limited insights into root causes and trends of errors	ML analysis provides valuable insights, helping organizations understand patterns and make informed decisions
Cost Efficiency	Manual error handling and troubleshooting can be resource-intensive	Automated error resolution through ML leads to cost savings by reducing manual intervention and downtime
Continuous Improvement	Difficulty in adapting error-handling strategies based on evolving system dynamics	ML models continuously learn and improve, ensuring error-handling mechanisms evolve alongside system changes

CEHS Anomaly Detection:

Efficient financial systems deemed to be effective should be effectual at detecting anomalies in the systems. Such systems are expected to search for careful mistakes before the errors worsen. Anomalies within the system are always unusual like a person who is keeping a keen eye on the road while driving. CEHS is useful in monitoring information aimed at the dynamics of how systems work, the type of transactions running, and resource utility. Established computer systems can be trained to learn traffic trends through the prism of numbers or computer techniques. Transience from typical rules like quick rise in error reports or elevated resource utility should be considered strange.

When a CEHS system finds an error, it can raise warnings to be used by engineers to prevent adverse outcomes. Big data analytics can be used in CEHS systems to link it with historical data or link it with other events that unfolded. This brings history to the table to help prevent such errors in the future. Consequently, test runs can be used in guessing events likely to occur in the future. Having a better grasp of the issue through the lens of CEHS aids in taking the right action thus giving warnings to staff involved in managing risks.

In some scenarios, CEHS slows traffic, changing transaction routes, or starting backup plans. This automatic reaction lowers time out and maintains the system's steadiness. The CEHS keeps getting better and adjusting. Over time, it becomes more skilled at finding differences that normal methods might miss. This helps spot possible problems with growing accuracy all the time. This plan-ahead way makes the CEHS strong in keeping big money systems safe from unexpected mistakes and problems.

ML Algorithm Selection:

Selecting machine learning algorithms circumspectly is central to CEHS's success. Carefully crafted algorithms ensure error detection, prediction, and the setting of anomalies are made distinct due to the algorithm's precision. The kind of system, possible mistakes, and what we want to achieve are important for choosing the best algorithms. For example, regression methods can be used to guess mistakes using old data. On the other hand, clustering techniques are great for finding odd things by seeing patterns in big piles of information. The tech group needs to inspect and try different methods. They must check how well they work and if they fit the one-of-a-kind features of their system.

Centralized Architecture Design:

To make the central structure of CEHS, you need to know all about its current parts and how they work. The design needs to allow easy joining with little interruption during regular activities. Setting up strong connections for quickly sharing live data between the CEHS and other system parts is very important. This ensures we get information all the time, so we can fix mistakes speedily and act quickly. Also, thinking about how things can grow and change is important. This is because the design has to fit more complicated situations when many transactions happen.

Centralized Error-Management using Machine Learning

Error Management in the Middle (CEM) is a big answer today. It's one place where all information about errors can be found together. The main part of this way is the Centralized Error Handling System (CEHS), which was made to simplify getting mistake files, check how errors happen, and start quick actions. The main structure of CEM offers some important benefits. These help make the system more dependable and work better.

One main good thing about CEM is that it makes things easier to see. The CEHS is a single place to get all data about mistakes [7]. It gives an overall picture of system problems. This complete view helps us better understand the main reasons and results of errors. Computer workers, like administrators and makers, learn about mistakes. This helps them make smart decisions to fix these problems and improve systems.

Another big plus for centralized error management is quicker responses. The grouped format of mistake data lets us make decisions together, making it faster to look into and respond to errors. By quickly finding and solving problems, CEM reduces the time systems are down. This also helps prevent major drops in how well these systems work. It is significant for the smooth working of busy systems that change often. If problems take too long to fix, it can make everything go wrong in a chain-like way.

Moreover, CEM makes sure all practices are the same throughout everything. We use the same process to handle errors all the time. This makes things more reliable and easier to keep up with. Standardization makes it easier to create and manage programs and helps teams work together [8]. This same way of handling mistakes lessens the chances for errors and helps create a steadier and more expected system environment.

Centralized Error Management is a big plan that not only deals with the issues of old-fashioned error handling but also brings many good things like better views, quicker reactions, and common ways to fix problems. By using CEM, businesses can make their systems stronger against mistakes. This will create a more reliable and fast computing area.

Moreover, the main structure needs ways to store data safely and quickly. This helps collect mistake reports and performance information altogether. This data storage place is the base for training machine learning models. It enables the system to learn and change based on past trends.

Machine Learning Integration

Putting Machine Learning into the big control system for fixing problems (CEHS) shows a new way of managing mistakes. This uses smart rules to improve how it handles them in many ways. This mixing adds a smart part that makes error-handling processes work much better and faster.

One main thing that Machine Learning (ML) does for the CEHS is guessing when people will make mistakes. Large language models used in artificial intelligence computing systems utilize data patterns to find errors and ensure surgical and precise responses. The straightforward strategy used in these algorithms enables engineers to take action to prevent issues from unfolding before time. When problems likely to occur are caught beforehand, it reduces unexpected outcomes, permitting firms to fine-tune how to use their resources to minimize damage and losses stemming from diverse doldrums.

Integrating machine learning approaches could assist in handling errors and easily find the primary cause of issues detected. Detecting the initial origin of the mistakes within the algorithm using complex systems is rather mundane but has been made easier using machine learning approaches. By looking at different data sets and linking them to various things, ML-improved CEHS can find the main cause of mistakes faster and more surely. This speeds up fixing problems so they're less likely to happen again.

Adding ML offers the chance to fix certain mistakes without human help. ML models can be taught to spot unique signs related to specific errors and make automatic repair actions independently [10]. This not only speeds up the fixing of mistakes but also lowers downtime and cuts down on needing help from people, boosting a stronger system for dealing with bugs.

Using Machine Learning in the CEHS improves how we handle mistakes. It adds looking ahead abilities, tells if anything strange happens, helps to find what's wrong faster than before, and may even fix issues by itself. This mix of high-tech tools not only deals with the problems of old mistake handling but also helps groups get ready for errors in a changing digital world.

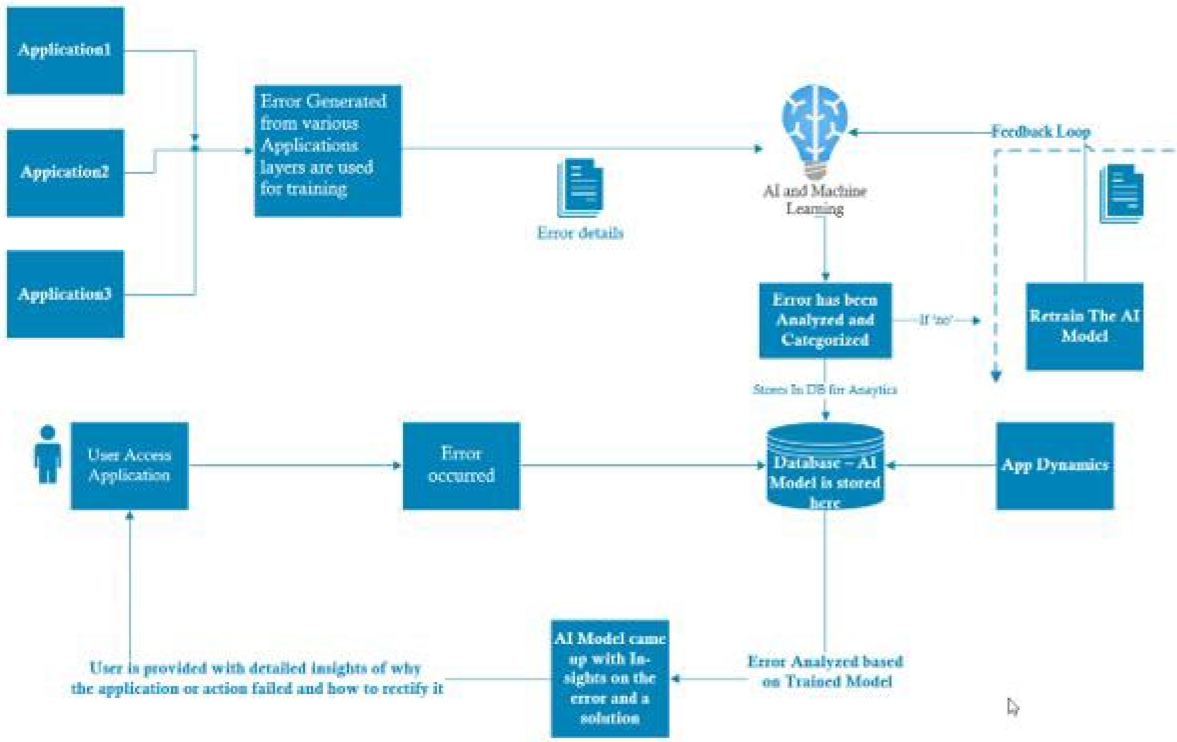

Figure -3

With new modern using AI and Machine Learning approach a model can be trained and built based on the historical data or a pre trained model can be used. The error detail historical data can be fed to the model training process and the resultant model can be deployed in server or a database where ML is enabled. Refer Fig-3. When an exception occurs, and error details are passed to trained model and the model will detect the message and predict the kind of exception that is happening and provide response that is user friendly. The model will also respond to the user with insights on how to resolve the error and move forward.

The Centralized Error Handling System (CEHS) needs a well-thought-out and step-by-step plan. It is important to use and connect the system smoothly with other existing techniques. We collected past mistake records and important system work info to start using it. They ensured the data they got was good and useful for teaching ML models. We did a first step to get the data ready. We fixed any mistakes or strange numbers that could make our training models inaccurate.

ML rules, picked carefully for the needs of mistake prediction and odd finding, were a big part of what was done. The choice-making focused on the kind of system, types of mistakes made, and goals wanted. The chosen models were made better using the ready data. A continuous improvement process was used to boost their performance [12]. This step-by-step process tried to improve at guessing mistakes and spotting weird things.

Connections were created to ensure the CEHS worked well with what was already there. These connections made it easy for the CEHS and other system parts to share information smoothly. This link was crucial for setting up quick error handling and responses, making CEHS a key part of the complete system design. A strong check-up system was implemented to always look at how well the CEHS is doing. The ML models were often tested to see how well they find mistakes and strange things. Changes were made to the models as needed, making sure they could change with time and conditions. Regular checking and review were crucial in keeping the CEHS working well in varying situations.

By doing these steps, groups put the CEHS in place. They used their skills to guess mistakes early and spot strange things with mistake-handling practices that got so much better overall. This smart joining not only handled current mistakes in error handling but also got the system ready to answer changing mistake situations quickly and wisely right away.

Potential Impact of CEHS using ML Algorithms

CEHS implemented using ML algorithms has some overarching advantages that are set to change how we handle mistakes and strengthen systems. Using active mistake prediction and automatic problem-solving tools in the CEHS is a big change for reducing computer system stop time. By guessing what might go wrong and quickly fixing it, the CEHS cuts down on time lost. This makes things run better and helps services be more available. This way of doing things before mistakes happen is a big change from old methods. Systems need to keep working without breaks or problems.

The CEHS can spot and deal with problems before they turn into big mistakes. This helps make systems more stable. The system gets stronger if we deal with possible issues before they happen. It can stay steady even when unexpected problems come up. This forward-thinking action matches with the fast pace of today's computer setups. It is significant to keep things running smoothly so we can always provide good services.

Gathering error logs in one place and using machine learning tools to find the cause of problems makes it easier to fix them. This not only makes it faster to find problems but also improves how we use resources to fix them efficiently. The CEHS can fix problems faster, which saves a lot of time and money. This makes errors have less effect on how well systems work.

The CEHS collects all types of data, uses ML techniques, and becomes a great source of valuable information about system behavior mistakes. These ideas give leaders the knowledge they need to make good decisions about improving systems and sharing resources [9]. The CEHS changes mistake data into useful knowledge, promoting a way to manage systems using information.

Challenges, Opportunities, and Future Research

As the CEHS promises big changes, it also brings out problems and new areas to look into in future studies. The use of ML machines can raise good questions about protecting data and safety. Strong ways to hide data and limit access are important for these problems. Keeping private information safe while using machine learning is a key part that needs ongoing focus in future study and action.

ML models' hidden nature and confusion make it hard to explain how they decide things. To get trust in the CEHS, we need to make clear and easy-to-understand models. System bosses need to understand and check what the CEHS does. This helps build confidence in how it runs. We should focus future research on making ML models easier to understand within the CEHS framework.

As systems change and transaction patterns shift, the CEHS must constantly keep changing. Online learning and active feedback loops are important ways to ensure that CEHS keeps doing its job over time [11]. Future studies should explore ways to change the CEHS smoothly as system behaviors shift. This will ensure it stays important and works well in changing work settings.

To get the most from CEHS, more studies could consider bringing it together with DevOps and non-stop integration/delivery systems (CI/CD). The link is instrumental in error handling, especially by permitting problem fixation and permitting updates. Changes will likely be made easier by linking errors to fast systems utilized in modern software development arrangements.

Conclusion

The Centralized Error Handling System (CEHS) is a big step forward in solving problems with errors within systems that handle lots of transactions. Previously, ML algorithms were carefully chosen, and a centralized layout was implemented for CEHS. This allowed it to lower how long something breaks down, improve system stability, and make fixing things easier. Examples from different areas like money and online shopping show how the system can deal with various wrong situations. But, problems like not keeping data safe and making it easy to understand in ML models show us why we need more study. The CEHS is ready to change and develop. Its future will make it even better at handling mistakes, helping systems that can bounce back easily in modern computing.

References

1. D.E.Frederick, "Deeplearningforlibrarians, "Library Hi Tech News,vol.37,no.1,pp.16−22,2020.
2. Saadoon, M., Hamid, S. H. A., Sofian, H., Altarturi, H. H., Azizul, Z. H., & Nasuha, N. (2022). Fault tolerance in big data storage and processing systems: A review on challenges and solutions. *Ain Shams Engineering Journal, 13*(2), 101538.
3. Kurani, A., Doshi, P., Vakharia, A., & Shah, M. (2023). A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting. *Annals of Data Science, 10*(1), 183-208.
4. Kend, M., & Nguyen, L. A. (2020). Big data analytics and other emerging technologies: the impact on the Australian audit and assurance profession. *Australian Accounting Review, 30*(4), 269-282.
5. Goodchild, M. (2021). Relational systems thinking: That's how change is going to come, from our earth mother. *Journal of Awareness-Based Systems Change, 1*(1), 75-103.
6. Shirani, F., Groves, C., Henwood, K., Pidgeon, N., & Roberts, E. (2020). 'I'm the smart meter': Perceptions of smart technology amongst vulnerable consumers. *Energy Policy, 144*, 111637.
7. Han, Y., Fan, L., Zheng, X. Z., Bai, J. M., & Han, Z. (2023). BayeSED-GALAXIES. I. Performance Test for Simultaneous Photometric Redshift and Stellar Population Parameter Estimation of Galaxies in the CSST Wide-field Multiband Imaging Survey. *The Astrophysical Journal Supplement Series, 269*(2), 39.
8. Shneiderman, B. (2020). Bridging the gap between ethics and practice: guidelines for reliable, safe, and trustworthy human-centered AI systems. *ACM Transactions on Interactive Intelligent Systems (TiiS), 10*(4), 1-31.
9. Yao, J., Crupi, A., Di Minin, A., & Zhang, X. (2020). Knowledge sharing and technological innovation capabilities of Chinese software SMEs. *Journal of knowledge management, 24*(3), 607-634.
10. Theissler, A., Pérez-Velázquez, J., Kettelgerdes, M., & Elger, G. (2021). Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry. *Reliability engineering & system safety, 215*, 107864.

11. Callan, G. L., DaVia Rubenstein, L., Barton, T., & Halterman, A. (2022). Enhancing motivation by developing cyclical self-regulated learning skills. *Theory Into Practice, 61*(1), 62-74.
12. Feng, D. C., Liu, Z. T., Wang, X. D., Chen, Y., Chang, J. Q., Wei, D. F., & Jiang, Z. M. (2020). Machine learning-based compressive strength prediction for concrete: An adaptive boosting approach. *Construction and Building Materials, 230*, 117000.