PROJECT NARRATIVE
1. COVER PAGE
Company Name: AI Pow LLC.
Project Title: Efficient Neural Architecture Search System to Manage and Analyze Complex Data.
Principal Investigador: Alfredo Costilla Reyes PhD.
Topic number/Subtopic Letter: 1a

2. PROPRIETARY DATA LEGEND – None

3. IDENTIFICATION AND SIGNIFICANCE OF THE PROBLEM OR OPPORTUNITY, AND TECHNICAL APPROACH

Identification and significance of the problem

The data produced at the Biological and Environmental Research (BER) program within the U.S. Department of Energy (DOE) Office of Science has increased by many orders of magnitude over the past few years. As reported in the workshop report on Grand Challenges from BER's Advisory Committee (BERAC), the challenges in data size and complexity, are continually intensifying with the adoption of new instruments and sensors that stream vast amounts of time-series data in real time [1]. The BER advisory council identifies the need for advanced technologies to integrate and analyze time-series data being generated through BER-supported community resources such as the ESS-DIVE [2] and WHONDRS [3] network. Particularly, anomaly detection systems are recognized as a valuable tool in large-scale complex time-series data because anomalies in data often point to areas that need urgent attention.

To better appreciate this problem, we examined a representative time-series ESS-DIVE data [4], which contains numerous daily data points from 2005 - 2016 to observe the intra and inter-annual variability and complexity of time-series data from 518 meteorological stations. Figure 1 illustrates the magnitude of complexity of the dataset by showing only evapotranspiration and precipitation variables for one station, which is only a fraction of the 518 datasets that provide six variables measured daily throughout 12 years.

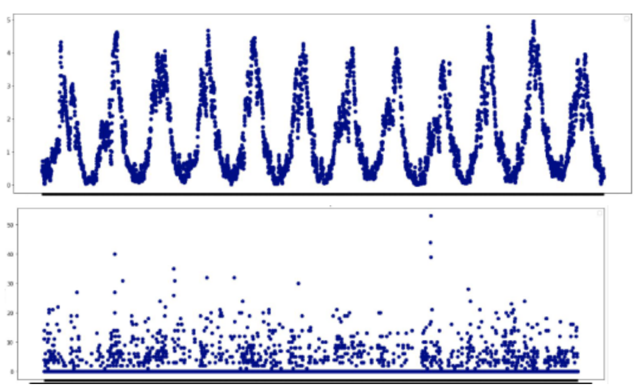

Fig. 1. Evapotranspiration (left) and precipitation (right) graphs of "*daily temperature and precipitation data for 518 Russian meteorological stations (1881 - 2010)*" in ESS-DIVE climate dataset [4].

After reaching out to community members at Pacific Northwest National Laboratory (PNNL) we have found that progress has been slow in integrating, analyzing data, and sharing key insights from multiple user facilities, community resources, instruments, and data systems. A particular need is that the DOE requires a robust, standardized, and cost-effective time-series outlier detection to identify anomalies in real-time data, which are visible symptoms of underlying problems that are relevant to engineers and scientists at the DOE. For example, Table 1 contrasts how different offices at the DOE have approached the time-series anomaly detection problem in complex data historically. Scientists who have monitored their data **manually** now find this method costly, poor to scale, and the slowest to detect anomalies when handling the ever-increasing amounts of data. Similarly, while hard-programmed **statistical** thresholds have improved scale and anomaly detection speed, their accuracy is still insufficient because such techniques can't easily 'self-adapt' to the ever-changing nature of complex-data.

The DOE is currently adopting complex business analytics tools to monitor their instruments and data systems automatically and promptly detect issues that **need immediate attention**. Machine Learning (ML) software has shown unprecedented results because of its ability to predict **anomalies and produce insights** in complex data beyond just detecting past errors. However, **current enterprise ML** still involves considerable computer science background and substantial budget sizes, not to mention that the output results are hardly understandable by other community of researchers due to its 'black-box' nature, limiting its usability and shareability inside the DOE.

Opportunity identification

AI Pow's proposed intuitive time-series monitoring platform has the potential to eliminate the engineering complexity and allow DOE's engineers and scientists to build and deploy advanced complex data analytics tools for their processes efficiently. The benefits from these advanced technologies include accurate anomaly detection, predictive instruments and data systems maintenance, and better understanding problems hidden in large amounts of complex data. But most importantly, the AI Pow's proposal aims to help DOE members, to quickly identify and understand anomalies in their datasets and take more reliable data-driven decisions.

We want to highlight that AI POW's sponsored research over the past 3 years with industrial partners, including General Motors, have allowed us to observe firsthand that this is a widespread problem in industry as process engineers have similar challenges to detect, interpret and resolve problems working with huge and disparate complex time-series datasets as the one presented in Fig. 1. Beyond this SBIR proposal, we also see a growing opportunity in creating better business monitoring tools as the manufacturing industry in the US alone already spends more than **$10 billion dollars yearly on suboptimal business analytics tools** [5]. Our value-proposition validation efforts at PNNL clearly show that by reducing the engineering complexity of complex-data analytic tools, we can accelerate its adoption as more domain experts are enabled to higher level of performance at substantially lower cost.

As shown in Fig. 2, our **proof-of-concept open-source system** [6] was used in our example ESS-DIVE climate dataset [4] to help researchers analyze multiple and unrelated datasets to identify one type of outlier detection algorithm (DeepLog) that determined the most important set of features that can be used to determine future anomalous precipitation events. Such information now reduces the amount of information to actionable items and, for example, help researcher focus on Solar Radiation as the main contributor when explaining past anomalous precipitation events.

While this example required deep computer science expertise to obtain the correct hyperparameter tunning and modeling the neural architecture, it helped us validate the need for a simpler and more affordable time-series outlier detection platform. The significance of the most complex project proposed in this SBIR proposal is that our proposed automation is tailored to dramatically extend our current ML techniques towards dealing with deep learning challenges raised in complex data that are large data volumes, variety, velocity, and veracity.

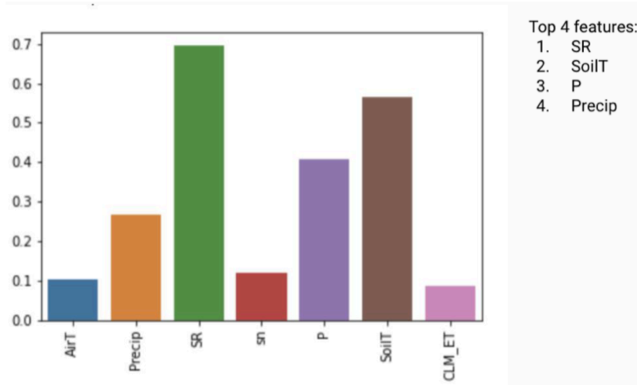

Fig. 2. Feature contribution (left) for the anomaly detection analysis in ESS-DIVE climate dataset [4]. Solar radiation (SR) is the most relevant feature to determine future anomalous precipitation.

AI Pow's AutoML proposal aims to help domain experts automate the complex data analysis task while intuitively understanding the origin of the anomalies to improve data analysis from multiple user facilities, community resources, instruments, and data systems. An automatic way to detect anomalous conditions in complex data will enable domain experts to have a general top view over the entire dataset and precision anomaly understanding per datapoint. Our proposed tool also offers domain experts the ability to visualize anomalies in context to ensure the best management judgment when finding fast and effective actionable tasks for time-sensitive problems. Table 1 highlights AI Pow's competitive advantages compared to the current solutions to manage complex data.

Comparison metric	Manual	Statistical	Current enterprise ML	**AI POW's AutoML**
Scale	**Low**	Medium	High	**High**
Anomaly detection speed	**Very Low**	High	High	**High**
Accuracy	Medium-high	**Low**	High	**High**
Cost of implementation	Low-medium	Medium	**High**	**Low**
Cost to maintain	**High**	Medium	Low	**Low**
Interpretability	High	Medium	**Low**	**High**

Table 1. Current complex data solutions compared to **AI Pow's** competitive advantages: ML automation, and interpretability.

Fellow collaborators at BER-supported communities have provided further detail on the importance of this endeavor to the DOE: "*I also want to express my support for the team's work on identifying and analyzing anomalies from different enterprise information systems. <u>It is crucial for the US to conduct research and development to deal with the complicated nature of anomalies in information systems. The anomaly detection task is of great interest for the different research communities at the Department of Energy and many other industry players</u>*." (Letter of support 1, Ms. Stephanie Burrier, **Pacific Northwest National Laboratory**).

In phase 1:

AI POW will investigate applications to 1) simplify the anomaly identification in time-series data while 2) allowing scientists to understand anomalies in complex data and 3) Lower the cost of adopting and sharing highly complex anomaly detection systems. **We will specifically focus on application areas 2, technologies and tools to integrate and analyze data from multiple user facilities, community resources, instruments and data systems, while positioning to address area 3 and expand datatype (tabular, images, video, etc.), anomaly detection at the edge (on-device), and database capabilities for sharing, mining and extracting knowledge from chemical and geochemical data in Phase II.**

Technical approach

AI Pow LLC will develop an AutoML system to perform data preprocessing, feature extraction, model selection, hyperparameter tuning, and prediction visualization, to manage and analyze extensive scientific data while enabling human interpretation to help subject matter experts to understand, trust and leverage advanced ML technologies. The resulting system will be a cost-effective and intuitive-to-use tool to accelerate the analysis, organization, retrieval, sharing, and modeling of complex data for the offices of Advanced Scientific Computing Research, Biological and Environmental Research, Basic Energy Sciences across multiple user facilities, community resources, instruments, and data systems which are goal of DOE topic 1a **applications 2.**

This AutoML project is based upon our team's proof-of-concept system AutoKeras [7], which has become the most popular, open-source AutoML systems (with over 8,100 stars and 1,300 forks on Github), which already has introduced significant advances in ML Automation and Interpretation applied to analyze complex scientific and engineering data sets. The proposed system is tailored to dramatically extend current ML techniques towards dealing with deep learning challenges raised in complex data that are large data volumes, variety, velocity, and veracity. Unlike independent data analytics algorithms, AI Pow will develop a flexible and modular end-to-end AutoML system to decide the optimal configuration of complex modeling pipelines automatically. This AutoML objective involves providing highly efficient data classification and regression (prediction) by systematically investigating two types of automation: pipeline searching and neural architecture search. For humans to understand the results generated by the AutoML system, AI Pow also proposes investing in AutoML interpretability to human-friendly explanations for subject-matter experts. Our innovative solution will allow domain experts to reduce bottlenecks and increase efficiency in managing and analyzing complex data for science and engineering using ML **regardless** of their computer science background.

Key technical challenges:
We anticipate three main technical challenges to achieve our goal: system design for further involving AutoML and interpretation, enabling automated machine learning for anomaly detection, and providing interpretations for automated anomaly detection system.

Challenges for interface design.
Building an end-to-end system to develop AutoML and interpretations is challenging and different from building a traditional end-to-end-system for a single customer. It requires a unified interface between various functionalities and the flexibility to support pipeline structure abstraction for automated machine learning and interpretation.

Challenges for enabling AutoML for outlier detection.
It is nontrivial to build an AutoML system for anomaly detection. First, it is difficult to define the search space. Unlike the search spaces defined by the existing AutoML systems, the search space of automated anomaly detection needs to cover various hyperparameters, preprocessing, and analysis modules. Second, the imbalanced nature of anomaly detection may introduce bias in the searching process and make the search easily fall into the local optima. Third, the search strategy needs to exploit the samples and historical search experiences more effectively since abnormal samples are sporadic.

Challenges for providing interpretations.
It is very challenging to provide automated anomaly detection systems to interpret automated anomaly detection systems. First, anomalies usually vary on the data; defining a set of unified features to give reasonable interpretation to the anomaly detection model requires extensive investigation. Second, different from traditional machine learning, AutoML employs a searcher to find the optimal model for the given data, and how to provide the interpretation for the searcher is critical to provide satisfactory interpretation to the users.

To tackle the challenges above, this project proposes to develope the following three core innovations:

1. **Highly modular end-to-end anomaly detection system**. Existing anomaly detection frameworks focus on finding a single machine learning model to detect the outliers. However, outlier detection is a case-specific task, and it is non-realistic to see all kinds of outliers using a single machine learning model since it typically requires different models even in the data collected from the same source but another period. Our system will implement various carefully selected components to provide all the necessary functionalities to construct a machine learning pipeline. The system architecture's flexibility will also enable us to develop advanced functionalities, including AutoML and interpretable ML.

2. **AutoML for anomaly detection**. To further save the engineering efforts and allow domain experts to build an anomaly detection system with limited machine learning knowledge, we will employ AutoML to find the best solution. To achieve this, we will equip the developed end-to-end system with three types of automation: automated pipeline searching for proper search space design, neural architecture search for automatically constructing optimal deep neural models, and active anomaly detection with meta-learning to leverage the label information efficiently.

3. **Interpretations for anomaly detection**. Apart from a well-performing solution, DOE scientists and engineers also demand interpretability. We propose to first offer interpretable machine learning solutions for users to understand the mechanism behind the pipeline's decision; then, enabling interpretations of AutoML for users to learn why the machine constructs such pipelines.

4. ANTICIPATED PUBLIC BENEFITS

ML is a disrupting force across many industries; Statista projects ML-solutions' demand to grow more than 50% year to year over the next five years. The proposed project well aligns with two of NSF's 10 Big Ideas to positively impact our community and society:

Harnessing the Data Revolution: As data science and engineering become more critical for businesses, it is paramount to consolidate a 21st-century data-capable workforce. Currently, many large enterprises can implement analysis detection systems because they can afford a highly specialized computer scientist team. This project aims to design easy-to-use tools tailored to software engineers with basic ML knowledge. This reduction in complexity and deployment cost can accelerate the technology adoption for a larger group of businesses that can now embrace the data revolution to start using their data assets to analyze complex data in new ways.

Future of Work at the Human-Technology Frontier: By increasing the adoption of the proposed advanced AI technologies, more companies and researchers could use complex anomaly detection frameworks. Successful human-technology partnerships could effectively enhance human performance to simplify excessive amounts of data into actionable items using current DOE's extensive datasets.

Complex data analysis solutions for the US manufacturing industry:
Anomaly detection software already is already used in industry to enable production supervisors to maintain high assembly reliability and quality control. The economic benefits of minimizing the time spent detecting and promptly resolving equipment issues range from product quality increase to production line downtime reduction. IBISWorld estimates that in 2020 alone, industrial equipment repair and maintenance generated a revenue of more than $13 billion dollars in the US [8].

AI Pow can lower the cost of ML-based anomaly detection tools. We want to highlight that business analysts at IBISWorld expect that anomaly analytics technology adoption for manufacturing will grow as more <u>medium and small manufacturers</u> can afford to invest in updating their computing infrastructure to capitalize on their data assets. Smaller firms are interested in generating an economic benefit from minimizing the significant waste in time and resources when they only can observe an abnormality in a mass-scale production process when measurable damage has already happened. This trend has originated from big manufacturing companies that are standardizing business analytics tools across its supply chain, such as the Volkswagen's industrial cloud [9].

5. TECHNICAL OBJECTIVES

In this Phase 1 program:

AI POW will develop a highly modular end-to-end anomaly detection prototype system that will (1) be tested on complex data at DOE. (2) integrate AutoML techniques tailored for DOE's complex data anomaly detection. (3) collect feedback from users at the DOE and Small and Medium manufacturers regarding the impact of the Interpretation features integrated in anomaly detection to measure its impact on efficiency and cost savings in their operations.

The Phase 1 activities will center on creating an end-to-end, automated, and interpretable anomaly detection system which can be readily adopted by our customers. Below are the key questions or objectives we will be focused on.

> **Objective 1:** Craft a highly modular end-to-end system. **Can we wrap commonly used outlier detection modules, including data processing, feature analysis, and detection algorithms, into primitives into unified interfaces?** The design of our proposed system's primitives will consider including data loading pipelines and scoring pipelines to truly enable an end-to-end outlier detection system from data to prediction results.
>
> **Objective 2:** Design an automated Outlier Detection. **Based on our end-to-end system, can we identify the most suitable pipeline effectively and efficiently for a given task?** The search space will consist of both primitives and hyperparameters. We will investigate different algorithms, including Bayesian optimization, reinforcement learning, and evolutional strategies.
>
> **Objective 3:** Demonstrate an Interpretable Outlier Detection. **Can we provide explanations of the predictions?** We will explore both intrinsic and post-hoc explanations. Besides, we will develop human-friendly interfaces to present the explanations to end-users. Please note that explainability is also meant to improve the access to learned models for broader use at the DOE.

Positive answers to the above will establish the path forward for a Phase II that can build to commercialize a broad reaching solution for DOE application area 3 to include non-uniform, unstructured and distributed data by developing multimodal approaches that encourage integration across types of data and even go a step further to integrate on-device technologies to analyze data.

6. WORK PLAN

The first part of this plan introduces relevant prior experience at AI POW to provide context for the expertise that will be applied to this project. The second part will detail the specific tasks proposed for Phase 1 and rigorously tie these to the objectives in section 7.

Related Research/R&D

The team has collaborated in DoD and industrial projects, co-authored publications, and worked together on active DARPA and NSF projects to conduct fundamental research in developing interpretable, automated, and interactive machine learning systems. Our past collaborations have provided a solid ground to potentiate the synergies emerging from continuous multidisciplinary interactions dedicated to the proposed data science-centered development.

In the work "A time-interleave-based power management system with maximum power extraction and health protection algorithm for multiple microbial fuel cells for internet of things smart nodes" [10], the PI worked with Microbial Fuel Cell (MFC) technology, a novel technology that can transform organic substrates in wastewater into electricity through a bioelectrochemical process. Such work with biological, microbial, and electrical processes are key to understand the type of data and goals for members of the Office of Biological and Environmental Research. Equally important is that this previous work also lays down a preamble to apply complex-data processing systems for IoT instruments that acquire time-series data in remote locations.

In "*Pyodds: An end-to-end outlier detection system with automated machine learning*" [11], the senior personnel of this proposal, presents an early prototype to optimize an outlier detection pipeline for a new data source at hand. Pyodds can define a limited search space in the outlier detection pipeline and produce a simple search strategy within the given search space. The proposed system here, builds upon this work to propose a more ambitions full-stack machine learning system for multivariate time-series data to perform three indispensable outlier detection scenarios: point-wise detection (time points as outliers), pattern-wise detection (subsequences as outliers), and system-wise detection (sets of time series as outliers).

In work "*Auto-keras: An efficient neural architecture search system*" [7]. Our team created the fundamental understanding to develop a robust Neural Architecture Search. While Auto-Keras doesn't support time-series analysis, in the current development of this work we propose novel framework enabling Bayesian optimization to guide the network morphism for efficient neural architecture search. Moreover, we are maintaining an open-source AutoML system based on our method, namely Auto-Keras that runs in parallel on CPU and GPU, with an adaptive search strategy for different GPU memory limits.

Finally, the work in "Technology enabling circuits and systems for the Internet-of-Things: An overview" [12] of our team is also relevant for this project. The Internet-of-Things (IoT) network is being vigorously pushed forward from many fronts in diverse research communities. However, many problems are still there to be solved. Particularly as a tremendous amount of data is generated by IoT equipment at the DOE, quick and intelligent data analysis where information is generated is vital to reduce the amount of some of that data without getting rid of useful information, giving rise to the integration of AI and IoT in the form of AIoT. This proposal also considers in Phase 1 the computational elements needed to enable new AIoT applications to manage complex data in Phase 2.

Phase 1: research approach and planned work
<u>Innovation 1: An End-to-End Anomaly Detection System</u>

Our primary research objective is to introduce end-to-end anomaly detection solutions, including data preprocessing, feature extraction, model selection, hyperparameter tuning, and prediction visualization. We have established two main milestones. First, by building on top of our previous anomaly detection package PyODDs, we will deliver an end-to-end system that supports various data loading modules, visualization modules, and essential model selection functions. Second, we will design and implement data preprocessing, feature extraction modules, and more anomaly detection algorithms to provide flexible pipeline construction.

Building an End-to-End System
This phase describes how we will build an end-to-end system that provides complete pipelines from data loading to prediction visualization. Our system will build upon PyODDS, our open-source anomaly detection system, with database support. We will first briefly introduce PyODDS and then describe how we will transform it into an enterprise-grade product.

PyODDS is an end-to-end Python system for Outlier Detection with Database Support. Figure 3 pictures an overview. In PyODDS, we have implemented 13 outlier detection algorithms, including traditional statistical approaches and more recent neural network frameworks. The implemented algorithms can deal with both static and time-series data. It supports operation and maintenance from TDengine, a light-weight SQL based database, which enables end-to-end database executions. PyODDS also supports model selection and hyperparameter tuning with Bayesian Optimization to automatically search for the optimal model for a given task. PyODDS is currently released under the MIT license and freely available on Github.

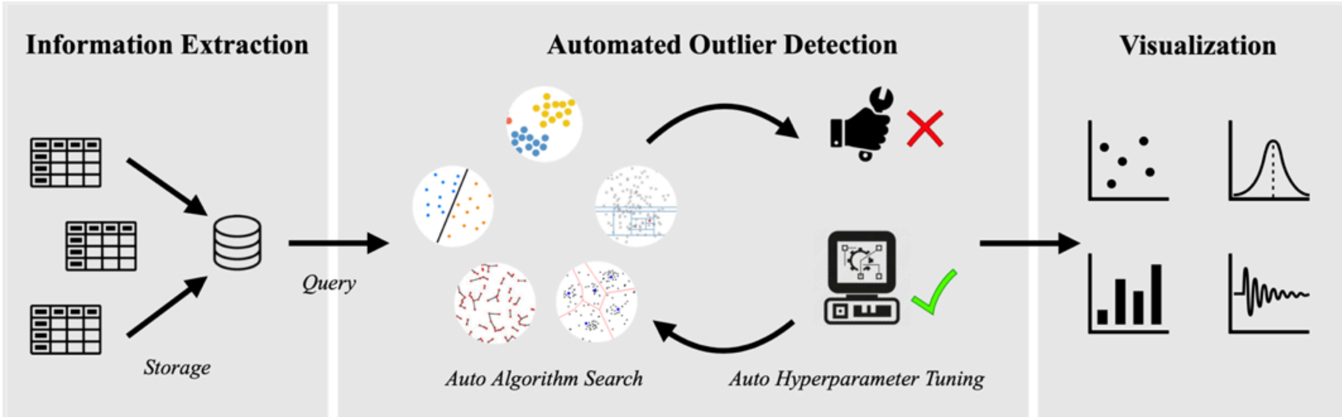

Figure. 3 an end-to-end anomaly detection system with algorithm search and hyperparameter tuning.

We will make several enhancements to PyODDS to allow its use as a commercial product. First, we will offer better database supports. In addition to TDengine, we will give unified database execution Application Programming Interface (API) for other mainstream databases, such as Apache Spark, Oracle, and MongoDB, so that assembly line managers can quickly deploy our product upon different databases. Second, we will provide an interactive Graphical User Interface (GUI) to allow users better understand and interpret the prediction. Specifically, we plan to develop our demo based on Plotly, an interactive front-end framework for machine learning, and Graphana, open-source analytics, and interactive visualization web application. Third, we will provide more anomaly detection algorithms and more efficient searching algorithms. We will implement more algorithms to deal with different data types, such as tabular data, images, and time series. We will also provide additional choices for searching algorithms based on reinforcement learning and evolutionary strategies.

Building a Flexible End-to-End System

We will build a highly flexible end-to-end system in this phase that provides an exhaustive machine learning process, including data loading, data processing, feature extraction, prediction, visualization analysis, and intuitive graphic user interface. Our system will build upon our experiences in the DARPA's D3M project and target time-series outlier detection, which has enormous potential detecting abnormalities in complex data applications, including fault detection, and malicious usage detection in various internet services. We aim to provide a transparent and extendable anomaly detection system with exhaustive functionalities, including data I/O, data processing, feature analysis, and detection algorithms, as well as an easy-to-use graphic user interface to allow human-AI interaction introduce domain-expertise. Here, we describe the design of our system.

In our proposed time-series outlier detection system (ODeSys), we propose to have mainly six modules: Data preprocessor, time-series processor, feature analyzer, a detection module, reinforcement module, and human-AI interface. Figure 4 illustrates the overall workflow and structural design of ODeSys. Specifically, each module consists of several primitive sets where each one is composed of various functions. This package aims to provide an end-to-end machine learning system for outlier detection tasks on time series data, and the target audience of this package is general software engineers with limited machine learning/data mining expertise. We define three scenarios that can include all of the outlier detection scenarios in our daily life, such as point-wise detection, pattern-wise detection, and system-wise detection. The point-wise detection aims at detecting the anomalous time point within the data by defining the anomalies as time points. The pattern-wise detection aims to identify odd patterns in the data by specifying the anomalies on subsequences. The system-wise detection aims to find anomalous systems by defining a set of time series data as an anomaly. Next, we illustrate the functionality of each module that composes ODeSys.

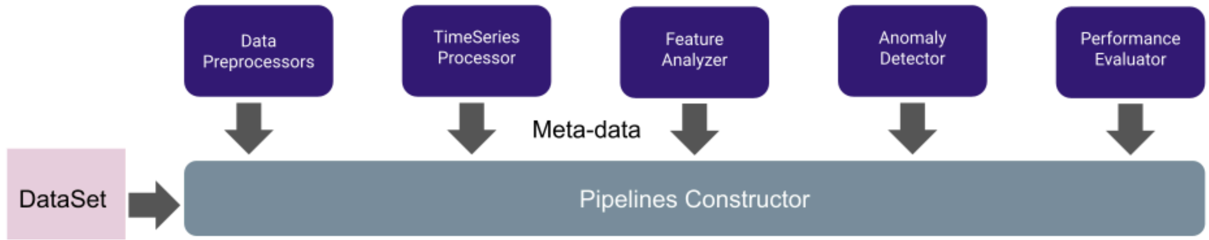

Fig. 4 ODeSys system overview.

Our team proposes the design the **data preprocessor** to be for general data preprocessing purposes, including loading data from various input sources (such as a database or multiple kinds of popular data types), data validation, missing data imputation, and data conversion. The design of the time series processor should tackle the unique attributes in the time-series data, such as trend and seasonality, by various smoothing and transformation techniques. **The feature analyzer** aims to help users to extract meaningful features from three aspects of time series data: temporal-domain [13] (e.g., statistical features), frequency domain [14] (e.g., spectral transformations), and latent feature space [15], [16] (e.g., matrix factorization). **The detection modules** will involve various popular and state-of-the-art machine learning algorithms for outlier detection, and we will categorize the algorithms based on the application scenario into each primitive set. For the detection modules, we will include several traditional point-wise approaches such as isolation-forest [17], local outlier factors [18], pattern-wise techniques such as matrix-profile [19], and state-of-the-art deep learning models including LSTM autoencoder [20], generative adversarial neural networks [21] as well as popular ensemble methods [22] to address system-wise detection scenario.

Typically, in the early stage, the outliers' labels are usually inaccessible; therefore, domain expertise plays a critical role in the beginning. **We propose the remaining two modules: human-AI interface and reinforcement module, to introduce human knowledge into the machine learning model**. The human-AI interface aims to provide a graphic user interface for users to manipulate and construct the outlier detection pipeline in a drag-and-drop fashion, visualizing the result and annotating the labels. On the other hand, the reinforcement module aims to incorporate the domain expertise via active learning and a rule-based system. The active learning approaches aim to efficiently use annotated labels to improve the existing machine learning pipeline, where the rule-based method will allow users to construct their rule-based model to incorporate their domain knowledge with a machine learning pipeline with a set of pre-defined rules.

Innovation 2: Automation in Anomaly Detection
We aim to provide an AutoML solution for anomaly detection to reduce human efforts in developing anomaly detection models. We will focus on three types of automation, including automated pipeline searching, neural architecture search, and active anomaly detection with meta-learning.

Automated Pipeline Search
We will apply AutoML techniques for pipeline search [23]. Given some candidate primitives, we aim to search for a pipeline that can achieve reasonably good performance on a given task. We will develop a tailored pipeline searcher by defining a search space for anomaly detection and optimizing the pipeline with Bayesian optimization and deep reinforcement learning.

We will divide the candidate primitives into four subsets for the search space: data preprocessing, time-series preprocessing, feature analysis, and detection algorithms. In each subset, we aim to search for a primitive or a combination of primitives to transform the data or make predictions. We will also search for the critical hyperparameters of each primitive to optimize the performance. For the searching algorithms, we will explore optimizing the pipeline through Bayesian optimization and deep reinforcement learning. In the beginning, we first randomly sample some pipelines. We train each pipeline on the training data and evaluate the pipeline's performance based on a subset of validation data. The performance on the validation data, i.e., whether it is improved or not, will serve as a rewarding signal to adjust the sampling strategy towards sampling pipelines that can deliver better performance.

Neural Architecture Search
Motivated by the success of deep learning in anomaly detection, we will explore automatically searching the optimal neural architectures [24]. The main challenge is how we can deal with imbalanced labels in anomaly detection data. Specifically, the anomalies are those rare instances that significantly deviate from the majority and contribute only a small portion of all the cases. The imbalanced nature will lead to unstable rewarding signals and make the search inefficient.

To tackle this challenge, we will implement a reinforcement learning framework based on our previous study of AutoOD [25]. We use an LSTM controller to sample architecture in each step and evaluate the sampled architecture's performance on validation data. The performance on validation data is used as a rewarding signal to update the controller. We devise a self-imitation learning module and a curiosity-guided search mechanism to encourage exploitation and exploration, respectively. To use the previous suitable architectures more effectively, we store them into an experience replay buffer. Those suitable architectures are sampled from the buffer periodically and update the controller with imitation learning. We use a curiosity-guided search to encourage the controller to test uncertain architecture to overcome local optimum to promote the search space exploration. We will wrap AutoOD as a primitive and support combined search of pipeline and neural architectures in our product.

Active Anomaly Detection with Meta-Learning
In real-world applications, anomaly detection algorithms tend to have high false-positive rates. In many real-world scenarios, analysts or domain experts will investigate the top instances one by one in a ranked list of anomalies identified by an anomaly detection system. This verification procedure generates informative labels that can be leveraged to re-rank the anomalies and help the analyst discover correct anomalies. Motivated by this, active learning is proposed for anomaly detection to discover more anomalies [26]. In our product, we will support a learning-based active anomaly detection with meta-learning [27].

Given some labeled anomaly detection datasets, we extract some meta-features, such as the output of an unsupervised detector and the distances to the currently observed normal/abnormal instances. We then train the meta-policy, a neural network, to explicitly model what instance the system should query next. The meta-policy is trained in a reinforcement learning fashion to optimize the cumulative discounted reward, i.e., the number of discovered anomalies in the long-term. Then the system can directly transfer the trained meta-policy to any unlabeled datasets for active anomaly discovery.

With enough data and fine-grained features, we expect the meta-policy to meta-learn the anomaly patterns in the active anomaly detection process and transfer to new datasets. An excellent property of this method is that it gives us the ability to apply the trained meta-policy network in our system without further tuning. In our proposed product, we plan to enable human-in-the-loop anomaly detection by pre-training a meta-policy to select the most promising instance for query automatically.

Innovation 3: Interpretation in Anomaly Detection
This research objective aims to provide interpretations for the built machine learning pipeline and deep learning models. We divide this goal into two phases: first, providing interpretable machine learning solutions to allow the user to understand how the machine makes the decisions based on constructed pipelines; second, enabling interpretations on AutoML to let users learn why does the algorithm build such pipelines to perform outlier detection on their datasets.

Outlier detection plays a crucial role in many industrial applications. A significant limitation of advanced outlier detection models is the lack of explainability. Typically, outliers could indicate the rarity of an observed pattern or a low probability that a given instance appears, but those outliers do not necessarily mean hazardous data points. Without providing rationale behind detection results, it is obscure to system developers or end-users why the results are derived, impeding users from trusting the models or taking further actions against the detected outliers.

To alleviate the problem, we have studied various interpretable machine learning models in multiple fields of application. In [23], the authors have proposed a one model-agnostic outlier interpretation method to address the outlier detection problem by resolving outliers' local context. Specifically, the outlier interpretation is defined with three aspects: abnormal attributes, outliers score, and the outlier's context clusters. Researchers have distilled interpretation from a series of classification tasks between normal class and outlier class. After interpretation, we have also shown that prior domain knowledge can be incorporated to adapt outlier detection results to different applications. Another explainable recommender systems framework was proposed in [29] to overcome the representation entangling problem in deep neural models. This work's main idea includes disentangling the interactions between latent representations in different neural layers, identifying multiple semantic factors from data, and dividing latent representations into segments according to their information source. To achieve the goal, we propose a novel neural architecture with graph convolution layers and segment the layers to force each layer to focus on different data aspects for interpretability. Although this work focuses on the recommendation system, we formulate the outlier detection problem as the recommendation system because the algorithm's goal is to recommend the most anomalous data point from a pool of data points.

The two previous works focus on what to provide as the interpretation and how to interpret the outlier detection model. Our goal is to elaborate on both projects and develop a unified interpretable outlier detection framework for novel neural models. Notably, we will follow the interpretation definitions of contextual outliers [28] to define and segment [29] the neural network layers and modify the learning objectives from item recommendation into outlier detection. The developed framework will also be implemented into our end-to-end systems to provide exhaustive interpretations for various scenarios of outlier detection.

Interpretable AutoML for Anomaly Detection
Given a well-defined search space and candidate primitives, we can leverage various optimization techniques such as Bayesian optimization and deep reinforcement learning to find the best combination from the candidates to construct an optimal pipeline for anomaly detection. However, the method mentioned above does not provide interpretation, and in some domains, it requires careful decision-making, such as complex data, interpretability is the bridge to connect the machine learning models and decision-makers. Two ways to provide the interpretation are: Tree-based searching framework [30] and knowledge graph reasoning [31].

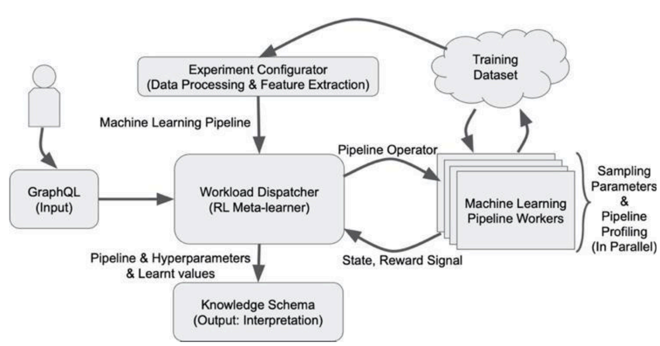

Fig. 5. An overview of explainable AutoML system with knowledge graph reasoning

The tree-based searching framework takes advantage of traceable paths in decision-tree algorithms to construct the tree-based pipelines from pre-defined search space with pipeline operators. Then, it employs a genetic algorithm [32] to iteratively optimize the pipeline. Another alternative [31] makes use of knowledge graph reasoning to provide interpretation in AutoML. Figure 5 illustrates the overview of the framework. In general, the framework leverages GraphQL to allow users to define the search space for the candidate pipeline. GraphQL provides a single API endpoint for data access, backed by a structured, hierarchical type system. Besides, it allows us to define a knowledge taxonomy to capture machine learning pipeline concepts, seamlessly populate facts to the predefined knowledge graph, and reason with them. The whole process of the framework is: first, defining the search space via GraphQL; second, leveraging reinforcement learning as a meta-learner to construct and test the candidate pipelines from the search space; third, output the knowledge schema of each searching iteration to provide the interpretation of the AutoML process.

The two systems mentioned above provide some interpretation level for automated machine learning, the interpretation's definition will be improved and clearly defined in Phase 1. We will formally define interpretation in the context of automated searching pipelines, and implement the system based on these two mentioned frameworks to alleviate the problem.

7. Statement of work

Phase 1 Statement of Work.
All work, including system design and experiments with data will be performed internally at AI Pow LLC headquarters in Texas.

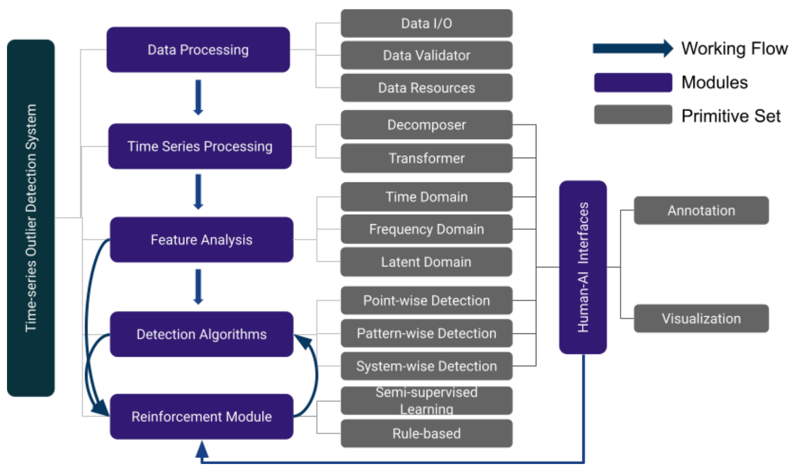

Figure 6. Building blocks of ODeSys: a fully integrated time-series outlier detection for complex data

<u>**Kickoff Meeting:**</u> AI POW will host a kickoff meeting with the Technical Point of Contact at DOE to discuss our approach. We will rally our team around objectives and seek out feedback early at kick off and weekly to mitigate or avoid risks to our critical deliverables.

Figure 6 is the blueprint for the building blocks of ODeSys, the time-series outlier detection system developed in task 1 and 2. Dr. Alfredo Costilla Reyes will perform or oversee its completion in the proposed timeframe.

<u>Task 1.</u> – Objectives 1 & 2. Developing End-to-End System (3 months): We will build upon our previous research and open-source efforts to develop an easy-to-use end-to-end machine learning system for time series outlier detection.

An end-to-end anomaly detection system. Existing anomaly detection frameworks focus on finding single machine learning model to detect the outliers, but outlier detection is a case-specific task and it is non-realistic to detect all kinds of outliers using single machine learning model since it typically requires different models even in the data collected from the same source but different period. Also, developing an optimal solution usually requires a machine learning pipeline including data preprocessing, feature extraction, model selection and post-processing. However, it is challenging to build an optimal machine learning pipeline for anomaly detection because there exists large amounts of pipeline components and selecting suitable components for anomaly detection is non-trivial. To address the problem, the PI, senior personnel and two programmers will build an end-to-end anomaly detection system upon our previous anomaly detection package PyODDs. Specifically, the system will implement various carefully selected components to provide the all functionalities required to construct a machine learning pipeline. The flexibility of the system architecture will also enable us to develop the advanced functionalities including AutoML and interpretable machine learning.

The main goals of this task are the creation of the building blocks of the End-to-End System. The main consideration is that the primitives should allow for a flexible End-to-End System. The six modules to develop are: Data preprocessor, time-series processor, feature analyzer, a detection algorithm module, reinforcement module, and human-AI interface.

<u>Task 2.</u> – Objectives 2 & 3. Developing Automated and Interpretable System (2 months): We will enable automated machine learning in our time series outlier detection. The team will focus on automatic pipeline discovery with default hyperparameters. Then we will develop algorithms to tune the hyperparameters

automatically. Finally, we will include interpretable outlier detection algorithms to provide explanations. We will test our AutoML product internally and run experiments on cloud services.

AutoML is not an easy task since wrongly defining the search space, and learning objective will lead to substantial computational costs, sub-optimal performances, and misusing the scarce label information. To reduce this risk, we will implement three automation types we have previously used in industrial projects to avoid this situation, including automated pipeline searching, neural architecture search, and active anomaly detection with meta-learning. The completion of the three computational blocks mentioned above will mark the completion of the end-to-end anomaly detection system. However, deep computer science background will still be needed to use the ODeSys.

AutoML for anomaly detection. To further save the engineering efforts and allow domain experts to build an anomaly detection system with limited machine learning knowledge, we will employ AutoML to find the best solution. However, employing AutoML is not an easy task since wrongly define the search space and learning objective will lead to huge computational costs, sub-optimal performances and misusing the scarce label information. To avoid this situation, we will conduct research on three types of automation, including automated pipeline searching for proper search space design, neural architecture search for automatically construct the optimal deep neural models, and active anomaly detection with meta-learning for efficiently leveraging the label information. Then, we will equip the developed end-to-end system with our research fruits on three types of automation. At this point a new programmer will be added to the team to start developing the front-end (Graphic User Interface) of ODeSys.

Interpretations for anomaly detection. Apart from well-performing solution, interpretability is usually required by high-stake domains such as healthcare and financial industries. Although interpretation for applications such as natural language processing and recommendation has been well-studied recently, the definition of interpretation for anomaly detection is still remain unknown due to its case-specific attribute, not to mention the interpretation for the AutoML. To address the problems above, we will conduct scientific research on interpretable outlier detection and AutoML, provide corresponding engineering solutions in our system. Specifically, we will first provide interpretable machine learning solutions for users to understand the mechanism behind the decisions made by the pipeline; then, enabling interpretations on AutoML for users to learn why does the machine construct such pipelines. We will develop the solutions into our system by following our previous research on explainable contextual outlier detection. Then, we will advance the end-to-end system by applying the tree-based algorithms and knowledge graph reasoning on interpretable AutoML for anomaly detection.

The completion of the Interpretable AutoML block for Anomaly Detection will mark the completion of Milestone 2. At this point ODeSys will be robust enough to be tested by domain experts at the DOE and potential partners working in the manufacturing industry. The beta release of ODeSys is programmed to be at month number 6 of this project and will be open to users by invitation-only.

<u>Task 3.</u> Polishing Product (2 months): We will open-source part of our codebase to allow end-users to try our AutoML system directly. We will collect some feedbacks and improve our product. In this period, our objective is to ensure we provide a human-friendly interface for end-users.

This task will deploy a beta version code, software, and demo datasets with the intention to evaluate user feedback and then learn from user testing results and iterate R&D development. This means that to ensure we are building the correct product to solve the needs of the DOE, we will further pair scientific R&D on interpretable outlier detection and AutoML, along with a build-measure-learn feedback loop to provide corresponding engineering solutions in our system to address the problems of topic 1a area of application 2. Specifically, we will expand our findings into our system by following our previous research on explainable contextual outlier detection following a lean methodology approach. We will then advance the end-to-end system by applying the tree-based algorithms and knowledge graph reasoning on interpretable AutoML for anomaly detection.

Task 4. Finalizing product (2 months): Based on the user's feedback, we will complete our work with improved interfaces. We will present our product in two options based on the input. First, we will tentatively wrap the product as software installed on personal computers or servers. Alternatively, we could deploy our development as a cloud service with a web-based user interface. We will choose one of the above options (or both) based on the user feedback from ODeSys' beta version release.

In Task 4 the PI along with Mr. Lai will collect evaluation results of system's experimental setup across the whole project and then analyze results for each product development iteration. This information will then be used to improve ODeSys and give directions to the programmer's effort. We want to highlight that an important part of this task will build user interfaces for each experimental implementation stage based on user interviews and technical point of contact at the DOE.

Task 5. Program Management and Reporting. AI Pow will comply with the report submissions as required by the DOE.

8. PERFORMANCE SCHEDULE

Overall, research and development will take nine months. We will start with system implementation and then focus on enabling AutoML and interpretation in our system. After that, we will spend half a year to polish and finalize our product based on the community's feedback. The milestone and timeline are listed as follows.

If this project is funded we plan to start its development on February 21, 2022 and the project will have a duration of nine months. During **Milestone 1** our team will generate the building block components of the End-to-End system proposed.
Milestone 2 will produce a beta version of the Anomaly Detection Automation and Interpretation system.
In **Milestone 3** our team will produce a more robust outlier detection system based on the data collected from the R&D deployment evaluation of each iteration of the system.
Milestone 4 will construct upon our proposed build-measure-learn feedback loop to provide the corresponding engineering solutions to advance our R&D innovation into a product offering.

PROJECT TIMELINE start February 21, 2022 - end November 21, 2022			Phase 1			Phase 2		Phase 3		Phase 4	
Research Objectives	**Tasks**	**Team**	M1	M2	M3	M4	M5	M6	M7	M8	M9
Milestones					M1		M2		M3		M4
Project starts	Kickoff meeting with the Technical Point of Contact at DOE	AC, XH, HL									
Weekly update	Weekly meeting with the Technical Point of Contact at DOE	AC, XH, HL									
O1: End-to-End Anomaly Detection System	Building an End-to-End System	AC, HL									
	Building a Flexible End-to-End System	AC, HL									
O2: Automation and Interpretation in Anomaly Detection	Automated Pipeline Searching	AC, HL									
	Neural Architecture Search	AC, XH, HL									
	Active Anomaly Detection with Meta-Learning	AC, XH									
	Interpretable AutoML for Anomaly Detection	AC, XH, HL									
O3: Continuous R&D innovation	Beta Release of Datasets, Code, and Software	AC									
	Evaluate Beta-Version Response of End-to-End system	AC									
	Learn From User Testing Results and Iterate R&D development	AC									
O4: R&D innovation advancement into a product offering	Collect Evaluation Results of Beta-Version Experimental Setup	AC, HL									
	Analyze results for next system development iteration	AC, HL									
	Build user interfaces for experimental implementation stage	AC									
Reporting requirements	Program management preparation and reporting	AC									

AC: Alfredo Costilla-Reyes
XH: Xia "Ben" Hu
HL: Kwei-Herng "Henry" Lai

M1 **End-to-End system**
M2 **Anomaly Detection Automation and Interpretation**
M3 **R&D deployment evaluation**
M4 **R&D innovation advancement into a product offering**

9. FACITIES/EQUIPMENT

The proposed work in Phase I will be carried out at AI POW LLC's headquarters located in Houston, Texas. In addition, the project can make use of the following assets.

AI POW LLC headquarters offer office space to perform R&D activities. AI POW LLC maintains its own secure network to facilitate research needs and to better support our computing facilities. Our Team also have individual access to essential Microsoft Office products.
Computing resources at AI Pow LLC: AI POW LLC currently source high performance computing resources from Amazon Web Services. Individual computing power available to AI POW LLC's R&D team

include equipment with the following characteristics: 2.3GHz 8-core Intel Core i9, Turbo Boost up to 4.8GHz, with 16MB shared L3 cache, 1TB SSD, AMD Radeon Pro 5500M with 4GB of GDDR6 memory and automatic graphics switching Intel UHD Graphics 630.

10. **RESEARCH INSTITUTION. None currently.**
11. **OTHER CONSULTANTS AND SUBCONTRACTORS.** There are no consultants or sub awardees expected to work on this project.
12. **PHASE II FUNDING COMMITMENT. None currently.**
13. **PHASE II FOLLOW ON FUNDING COMMITMENT. – None currently.**
14. **BIBLIOGRAPHY & REFERENCES CITED**
15. **References**

[1] G. 'Stacey, "Grand Challenges for Biological and Environmental Research: Progress and Future Vision. A Report from the Biological and Environmental Research Advisory Committee," Jun. 2017.

[2] O. of S. U.S. Department of Energy, "Environmental Systems Science Data Infrastructure for a Virtual Ecosystem (ESS-DIVE)," *http://ess-dive.lbl.gov/*, Oct. 05, 2021.

[3] P. N. N. L. U.S. Department of Energy, "Worldwide Hydrobiogeochemical Observation Network for Dynamic River Systems (WHONDRS) Network," *https://www.pnnl.gov/projects/WHONDRS*, Oct. 08, 2021.

[4] O. Bulygina and V. Razuvaev, "Daily Temperature and Precipitation Data for 518 Russian Meteorological Stations (1881-2010)}," Jan. 2012.

[5] D. Cook, "IBISWorld US Industry (NAICS) Report 51121C. Business Analytics & Enterprise Software Publishing in the US," 2020.

[6] Y. Li, D. Zha, P. Venugopal, N. Zou, and X. Hu, "PyODDS: An End-to-end Outlier Detection System with Automated Machine Learning," in *Companion Proceedings of the Web Conference 2020*, 2020, pp. 153–157.

[7] H. Jin, Q. Song, and X. Hu, "Auto-keras: An efficient neural architecture search system," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1946–1956.

[8] G. Holcomb, "IBISWorld US Industry (NAICS) Report 81131. Machinery Maintenance & Heavy Equipment Repair Services in the US," 2020.

[9] G. Roberts, "Siemens named as VW cloud integration partner.," *Aroq - Just-Auto.com (Global News)*, 2019, Accessed: Oct. 09, 2021. [Online]. Available: https://search.ebscohost.com/login.aspx?direct=true&db=bsu&AN=136276164&site=eds-live

[10] A. Costilla Reyes, C. Erbay, S. Carreon-Bautista, A. Han, and E. Sánchez-Sinencio, "A Time-Interleave-Based Power Management System with Maximum Power Extraction and Health Protection Algorithm for Multiple Microbial Fuel Cells for Internet of Things Smart Nodes," *Applied Sciences*, vol. 8, no. 12, 2018, doi: 10.3390/app8122404.

[11] Y. Li, D. Zha, P. Venugopal, N. Zou, and X. Hu, "PyODDS: An End-to-End Outlier Detection System with Automated Machine Learning," in *Companion Proceedings of the Web Conference 2020*, 2020, pp. 153–157. doi: 10.1145/3366424.3383530.

[12] J. J. Estrada-López, A. Abuellil, A. Costilla-Reyes, and E. Sánchez-Sinencio, "Technology Enabling Circuits and Systems for the Internet-of-Things: An Overview," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018, pp. 1–5. doi: 10.1109/ISCAS.2018.8351876.

[13] M. Barandas *et al.*, "TSFEL: Time Series Feature Extraction Library," *SoftwareX*, vol. 11, p. 100456, 2020.

[14] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *2007 IEEE Conference on computer vision and pattern recognition*, 2007, pp. 1–8.

[15] J. Mei, Y. de Castro, Y. Goude, and G. Hébrail, "Nonnegative matrix factorization for time series recovery from a few temporal aggregates," in *International Conference on Machine Learning*, 2017, pp. 2382–2390.

[16] H.-F. Yu, N. Rao, and I. S. Dhillon, "Temporal regularized matrix factorization for high-dimensional time series prediction," in *Advances in neural information processing systems*, 2016, pp. 847–855.

[17] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 413–422.

[18] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.

[19] E. Keogh, J. Lin, and A. Fu, “Hot sax: Efficiently finding the most unusual time series subsequence,” in *Fifth IEEE International Conference on Data Mining (ICDM’05)*, 2005, pp. 8–pp.
[20] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, “LSTM-based encoder-decoder for multi-sensor anomaly detection,” *arXiv preprint arXiv:1607.00148*, 2016.
[21] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, “MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks,” in *International Conference on Artificial Neural Networks*, 2019, pp. 703–716.
[22] Y. Zhao and M. K. Hryniewicki, “DCSO: dynamic combination of detector scores for outlier ensembles,” *arXiv preprint arXiv:1911.10418*, 2019.
[23] I. Drori *et al.*, “AlphaD3M: Machine learning pipeline synthesis,” 2018.
[24] B. Zoph and Q. v Le, “Neural architecture search with reinforcement learning,” *arXiv preprint arXiv:1611.01578*, 2016.
[25] Y. Li *et al.*, “AutoOD: Automated Outlier Detection via Curiosity-guided Search and Self-imitation Learning,” *arXiv preprint arXiv:2006.11321*, 2020.
[26] H. Lamba and L. Akoglu, “Learning On-the-Job to Re-rank Anomalies from Top-1 Feedback,” 2019.
[27] D. Zha, K.-H. Lai, M. Wan, and X. Hu, “Meta-AAD: Active Anomaly Detection with Deep Reinforcement Learning,” *arXiv preprint arXiv:2009.07415*, 2020.
[28] N. Liu, D. Shin, and X. Hu, “Contextual outlier interpretation,” *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.
[29] N. Liu, Y. Ge, L. Li, X. Hu, R. Chen, and S.-H. Choi, “Explainable Recommender Systems via Resolving Learning Representations,” *arXiv preprint arXiv:2008.09316*, 2020.
[30] R. S. Olson and J. H. Moore, “TPOT: A tree-based pipeline optimization tool for automating machine learning,” in *Workshop on automatic machine learning*, 2016, pp. 66–74.
[31] A. Elkholy, F. Yang, and S. Gustafson, “Interpretable Automated Machine Learning in Maana (TM) Knowledge Platform,” *arXiv preprint arXiv:1905.02168*, 2019.
[32] W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone, *Genetic programming*. 1998.