applied sciences

MDPI

Article

Big Data Mining and Classification of Intelligent Material Science Data Using Machine Learning

Swetha Chittam [1], Balakrishna Gokaraju [2,*], Zhigang Xu [3], Jagannathan Sankar [3,*] and Kaushik Roy [1,*]

[1] Department of Computer Science, College of Engineering, North Carolina A&T University, 1601 E. Market Street, Greensboro, NC 27411, USA; schittam@aggies.ncat.edu
[2] Engineering Research Center & Center for Visualization and Computation Advancing Research (ViCAR), Department of Computational Data Science and Engineering, College of Engineering, North Carolina A&T University, 1601 E. Market Street, Greensboro, NC 27411, USA
[3] Department of Mechanical Engineering & Engineering Research Center, College of Engineering, North Carolina A&T University, 1601 E. Market Street, Greensboro, NC 27411, USA; zhigang@ncat.edu
* Correspondence: bgokaraju@ncat.edu (B.G.); sankar@ncat.edu (J.S.); kroy@ncat.edu (K.R.)

Abstract: There is a high need for a big data repository for material compositions and their derived analytics of metal strength, in the material science community. Currently, many researchers maintain their own excel sheets, prepared manually by their team by tabulating the experimental data collected from scientific journals, and analyzing the data by performing manual calculations using formulas to determine the strength of the material. In this study, we propose a big data storage for material science data and its processing parameters information to address the laborious process of data tabulation from scientific articles, data mining techniques to retrieve the information from databases to perform big data analytics, and a machine learning prediction model to determine material strength insights. Three models are proposed based on Logistic regression, Support vector Machine SVM and Random Forest Algorithms. These models are trained and tested using a 10-fold cross validation approach. The Random Forest classification model performed better on the independent dataset, with 87% accuracy in comparison to Logistic regression and SVM with 72% and 78%, respectively.

Keywords: data mining; mongodb; No-SQL database; classification algorithms; logistic regression; support vector machine SVM; random forest

Citation: Chittam, S.; Gokaraju, B.; Xu, Z.; Sankar, J.; Roy, K. Big Data Mining and Classification of Intelligent Material Science Data Using Machine Learning. *Appl. Sci.* **2021**, *11*, 8596. https://doi.org/10.3390/app11188596

Academic Editors: Carsten Felden, Nicola Castellano, Henning Baars, Bruno Maria Franceschetti, Michiyasu Nakajima, Fanny-Eve Bordeleau and Trinks Sebastian

Received: 11 August 2021
Accepted: 9 September 2021
Published: 16 September 2021

1. Introduction

Our study is based on Intelligent Material science data (Magnesium-Alloy, Mg-alloy), which is structured to be a lightweight alloy without compromising the strength of the material and is widely used across multiple domains, including the United States Armed Forces [1–3]. The Mg-alloy data is composed of different metal compositions and their processing parameter information involving different techniques such as casting, extrusion, rolling, forging [4–11], etc. Currently, there is no existing data repository for storing Mg-alloy data. In this study, we propose a big data repository that archives the Mg-alloy data retrieved from scientific journals on which big data analytics, predictive modeling can be performed. The Mg-alloy tests are classified based on tensile strength properties including strength and the ductility of the metals. Based on these tensile properties, we propose the prediction model that will help in the future to determine the boundary conditions for strong Mg-alloy material for the specific purposes of an application.

We will focus on how to address this material science research problem in creating a big data repository and predicting the strength of the material by applying Machine Learning (ML) models to the big data retrieved from the data repository. We have applied MongoDB for management of a Mg-alloy big database and a logistic regression algorithm for its binary classification [12–14]. MongoDB, a big data tool, for data management of Mg-alloy is chosen, as NoSQL (non-SQL, non-Structured Query Language or non-relational)

databases like MongoDB are often better suited for storing and modeling semi-structured Mg-alloy data [15–17]. The Mg-alloy data is stored as documents on MongoDB, which is a popular NoSQL database [13]. The documents are queried from the Mongo collection to retrieve the tensile properties of the Mg-alloy as these properties of metals will help us to determine the strength of metals.

Tensile is a mechanical testing method, through which tensile properties such as tensile yield strength (YS), ultimate tensile strength (UTS), and elongation-at-fracture (EL) of the material under testing can be obtained. These three tensile properties are independent of each other. Ductility of the metal depends on the tensile elongation-at-fracture (EL). The higher the elongation, the higher the ductility, and vice versa. Our goal is to find a statistical correlation between these three independent variables using machine learning (ML) predictions. Using ML techniques, ductility (a physical property) of the Mg-alloy is determined by considering the tensile yield strength (YS), ultimate tensile strength (UTS) as independent variables.

Mg-alloy can form a strong or weak material, based on the combination of metal composition, its processing methods/parameters and tensile properties. In our methodology we are only considering tensile properties due to the sparse nature of the dataset in the variables of metal compositions, processing methods/parameters. Strong Mg-alloy is ductile metal and weak Mg-alloy is brittle metal. Mg-alloy can belong to either of these two classes. To address this classification problem, there are many possible popular Machine Learning classification algorithms, such as Support vector machine (SVM), Linear regression, K-nearest neighbor, Logistic regression, Random Forest, etc. [18–24]. However, choosing the best algorithm to solve a given problem depends on the number of factors; namely, accuracy of the model, data set formats, number of parameters, features, etc. Further, based on the above factors, nature of problem, presence of sparse dataset and the number of dependent and independent variables, the Logistic regression algorithm was chosen for Mg-alloy classification.

The classification algorithms Support vector machine SVM and Random Forest are also applied on the independent variables to predict the output variable, ductility of the metal. The comparison of classification metrics of all the three algorithms is done.

We trained our model on the extracted dataset with 128 data rows, tested on 33 data rows of independent dataset and observed from our results that we achieved significant accuracies greater than 70% for all the three machine learning classification models—Logistic regression, SVM and Random Forest. The accuracies using 10-fold cross validation on the train dataset of 128 data rows for Logistic Regression, Support Vector Machine SVM and Random Forest are 70.25%, 75.76% and 72.69%, respectively, and the accuracies when tested on independent test data of 33 rows were 72.72%, 78.78 and 87.87%, respectively.

2. Literature Review

The Mg-alloy is used in a variety of applications due to its abundant availability [2,3,25–32]. It has numerous advantages when compared to other metals. Mg-alloy is a light-weight metal with high ductility when processed using the right techniques and at appropriate conditions [1]. Because of its enormous applications and benefits, there are numerous ongoing research studies in the material science field to determine the composition of different metals to form a high-quality Mg-alloy.

The Magnesium-alloy data is collected manually in tabulated logbooks from scientific literature review and laboratory experiments [33–36]. Hence, there is a high need of developing big material science data repositories to automate the strong Mg-alloy metal based on material characterization, process parameters/methods. This leads to the proposal of the big data repository for the Mg-alloy data. Unless there is a big data repository, we will not be able to apply automatic machine learning (ML) predictions to determine the strength of Mg-alloy product. The selection of the database is based on different factors under consideration such as domain, structured, unstructured data, relation between data elements, data size, complexity, cost, scalability, and data analytics tools, etc.

As the Mg-alloy data is semi-structured data, the traditional relational database management systems (RDBMS), SQL (Structured Query Language or relational) databases may lead to horizontal scalability issues for storing the dynamically changing semi-structured data [15].The RDBMS cannot store or process the big data efficiently and the emerge of No-SQL database(non-SQL or non-relational) has become the popular solution to overcome this issue [37]. The No-SQL databases are better suited for semi-structured or unstructured data and outperforms in comparison to RDBMS [38]. When exponentially increasing big data was stored on RDBMS, it was observed that the performance of the database was very low while applying data extraction techniques, due to SQL query execution times being slower for these operations [37]. These performance issues can be addressed with No-SQL databases by expanding the horizontal scalability [39]. To overcome these efficacy problems, different types of No-SQL databases emerged, and one of those is the popular Mongo-DB which supports a document data model [40]. In a recent study, the Electronic Health Records (EHR) data is semi-structured big data and MongoDB is used for data management and analysis of EHR data [15]. Similarly, the Mg-alloy data is semi-structured data and can be efficiently stored on a No-SQL database.

Mg-alloy data is semi-structured big data that grows at an ever-increasing rate, once the data collection process is automated, and a data repository is created. Thus, we choose MongoDB as the big data repository for the Mg-alloy data, which is suitable for semi-structured data and has a cost of one-tenth when compared to RDBMS [15]. The possible data models for the No-SQL databases are key-value, column-oriented or document database [41].The data on MongoDB is stored as a document, and the document contains key-value pairs to hold the data [13,41]. MongoDB also supports embedded data models which reduces I/O operations [13]. It provides the redundant data or replica sets distributed across the cluster [15]. The 112 datasets are built from a larger body of literature to collect different metal compositions of Zn, Al, Mn, Ca, Si and their processing parameters information. Only rolled and extruded alloys are used to build the dataset by including the attributes related to extrusion [42].

In another logistic regression study, the authors have integrated conjugate gradient method to the logistic regression model for addressing the binary classification of customer churn problem for airline business [43]. A machine learning based spatio-temporal data mining approach is used to detect the HABs (Harmful Algal blooms) events in the region of Gulf of Mexico [20,23]. Here, the authors used the Kernel based support vector machine as a classifier in the detection of HABs and also predicting within a window of 7 days [20,23]. In one of the recent studies, Logistic regression is used not only to predict the football matches' results, but also to determine the significant variables that contribute to win a match [12]. In [44], the ML based random forest classifier is used for the fault classification in the power systems network. The comparative study of ML techniques, such as random forest, SVM, KNN and Logistic regression techniques has been explored for fault detection [44].

The ML models can also be used for detecting cyber-attacks. It is also equally important for the ML model itself to be robust against the adversarial examples to avoid misclassification or incorrect prediction. An adversarial example is a sample created by adding a little noise to the original sample data, although presenting no change identifiable to human perception will be misclassified by a deep neural network [45]. In recent studies, the machine learning security against adversarial examples is implemented for textual classification [45–48].The syntactically controlled paraphrase networks (SCPN) are developed to generate the adversarial examples for both sentiment analysis and textual entailment [46].These examples are introduced in the training data to increase the robustness of these models to syntactic variation.

In [47], the authors discovered that the ensemble of the weak defenses is not sufficient to provide strong defense against adversarial examples. A friend-guard adversarial example is created that will be correctly classified by the friend model and misclassified by the enemy model without introducing any changes in meaning or grammar that will be perceived by humans [47]. The machine learning security is not only used for generating

the adversarial examples but also used to detect the backdoor attacks for multiple Deep neural networks [49].

In one of the previous Mg-alloy studies, the popular machine learning models Support vector machine SVM, Artificial neural network ANN is used to predict the mechanical properties of the Mg-alloy [18–24,42]. The tensile properties YS, UTS and Elongation-at-fracture (EL) are the three outputs of the output layer [42]. We are using mechanical properties of the alloy YS, UTS as the input variables, to predict the output variable ductility which depends on tensile elongation-at-fracture (EL). We have applied machine learning predictions of three popular classification algorithms—Logistic regression, Support Vector Machine SVM and Random Forest on the Mg-alloy data and our contributions to this study are:

- Created a big data storage for material science data.
- Developed data mining techniques to retrieve the required data from a database to perform data analytics.
- Developed machine learning prediction model to determine the strength of metals.

3. Methodology

The methodology includes three steps of implementation; namely, Data Management, Big Data Mining and Data Classification. Each step is discussed in detail.

3.1. Steps of Implementation

3.1.1. Data Management

Listed below are the steps that we followed for archival of Mg-alloy big data onto the MongoDB. These steps include the details of the processes starting from Data collection to the Data storage onto the MONGODB.

1. Data Collection
2. Data Cleaning/Data Preprocessing
3. Data Model Design
4. Data Conversion Model
5. Data Transfer Model

Here, steps 1 and 2 are done manually for collecting and preprocessing the data, and detailed information is discussed in Sections 4.1 and 4.3, respectively. For steps 3 and 4, we developed the code using python programming language. This program includes both the functionality of the data grouping as per the schema design chosen, and conversion of input CSV (Comma Separated Value) file format to a JSON (JavaScript Object Notation) file format. The technical information of steps 3 and 4 is discussed in detail in Sections 4.2 and 4.4, respectively.

We observed that the python program developed combinedly for step-3 and 4 took the execution time ranging from 48–68 ms for multiple attempts, to achieve the grouping of the input data of 218 rows per the data model design, and converting it into JSON file format resulting in 143 documents, with embedded document structure. For step 5, we used Studio 3T, a Mongo GUI (Graphical User Interface) tool to import the data onto the database. When we tested the data import functionality with multiple execution attempts, it was observed that the execution times ranged from 83–152 ms for importing 143 documents on to the database.

The statistics for the execution times in steps 3 to 5 are listed in Table 1.

Table 1. Execution times for the functionality Data model design, conversion and transfer model.

Steps	Functionality	Input File Size	Output File Size	Execution Times in Milliseconds (ms)	
Step-3, 4	Data model design & Data conversion model	110 KB	516 KB	Attempt-1	53 ms
				Attempt-2	48 ms
				Attempt-3	68 ms
Step-5	Data Transfer Model	516 KB	345 KB	Attempt-1	152 ms
				Attempt-2	124 ms
				Attempt-3	83 ms

3.1.2. Big Data Mining

Data mining technique includes the below two steps. These steps are implemented for retrieval of Mg-alloy tensile properties from the database in JSON file format and parsed to a CSV file format.

1. Query document to retrieve tensile properties of metals resulting in JSON file format.
2. Convert JSON to CSV file format and prepare the dataset to feed the machine learning model.

We developed two programs using python programming language for the above two steps. The technical details are discussed in Section 5. The two programs are:

- Program-1 is for retrieving all existing tensile values from database and parsing the result to the CSV file format to feed the machine learning ready.
- Program-2 is for retrieving all numerical tensile values from the database and parsing the result to CSV file format to feed the machine learning ready.

When programs 1 and 2 were run multiple times, we noticed that the execution times for combined functionality of data retrieval from MongoDB and parsing the result to CSV file format ranged from 270–554 ms.

The statistics for the execution times for Program-1 and 2 for multiple execution attempts are listed in Table 2.

Table 2. Execution times for data retrieval from database and parsing the resultant JSON file to CSV file format.

Program	Functionality	Output File Size	Execution Times in Milliseconds (ms)	
Program-1	Retrieve all existing tensile values and parse result to CSV file format	10 KB	Attempt-1	499 ms
			Attempt-2	387 ms
			Attempt-3	554 ms
Program-2	Retrieve all numerical tensile values and parse result to CSV file format	8 KB	Attempt-1	270 ms
			Attempt-2	309 ms
			Attempt-3	301 ms

3.1.3. Data Classification

The Logistic regression model is built with tensile properties of metals as input and output variables. Tensile Yield Strength (YS) and Ultimate Tensile Strength (UTS) are used as input variables to predict the output variable ductility which depends on Elongation-at-fracture (EL).

3.2. Model Diagram

Figure 1 is the model diagram representing the flow of the processes involving data management, big data mining and data classification.

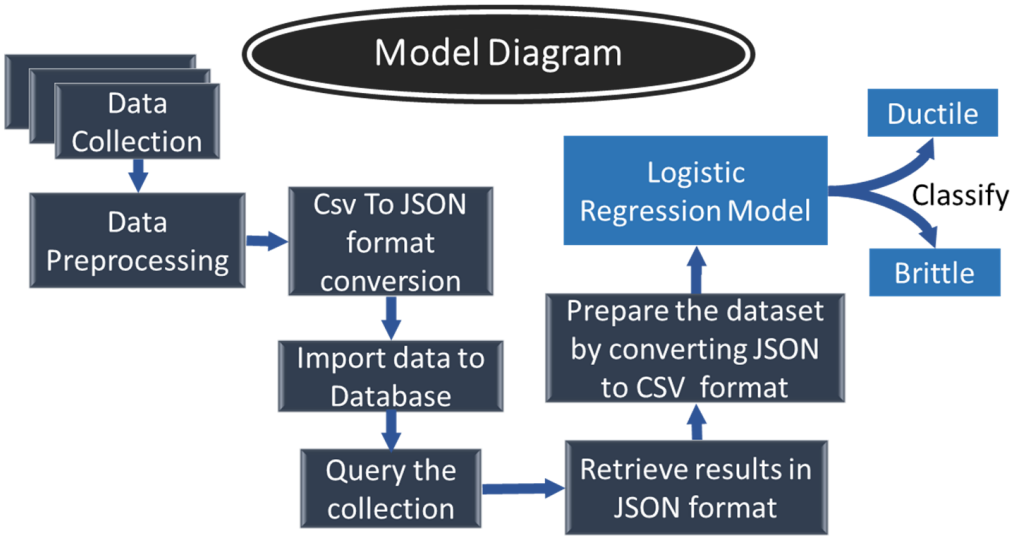

Figure 1. Model Diagram for the classification of Material Science data.

4. Data Management

4.1. Data Collection

The Mg-alloy data was collected from scientific articles through the literature review [33–36]. The retrieved data was tabulated in a semi-structured CSV file format. The information in the data collected includes the details of the author, publisher, followed by Mg-alloy details. The Mg-alloy details include the metal compositions, its processing parameters, mechanical and corrosion properties. The processing information when further drilled down includes the details of casting, extrusion and rolling temperatures of the alloy and the mechanical properties includes the tensile and compression properties of the Mg-alloy. The data collected from each scientific article may have multiple combinations of the metal compositions, its processing, mechanical and corrosion properties to form magnesium-alloy. Each material composition combination is written as a single row of data in the CSV file. Multiple scenarios exist where multiple rows of data belong to one single scientific journal.

The data is raw and was cleaned and preprocessed manually. This cleaned data was then exported to the database. As the clean data was in a semi-structured CSV file, the format needs to be converted to a JSON so that it can be imported onto the No-SQL database.

4.2. Data Model Design

MongoDB supports two types of data model design, such as the embedded data model and the normalized data model [13]. The embedded data model is further classified as a one-to-one relationship embedded document and one-to-many relationship embedded document. In general, normalized data design is used for complex many-to-many relationships between the connected data. So, here the decision of the data design and whether to choose the embedded model or normalized model depends on how the data is connected and what kind of relation is formed between the data. When we have analyzed the Mg-alloy data manually, we observed that for most of the articles, there are multiple data rows related to each scientific article and only few data rows are holding one-to-one relation. So, with most of the one-to-many relation between data, we have decided to choose one-to-many embedded model design for the Mg-alloy data. The reason for choosing the embedded models is, in each row of the Mg-alloy data, the material compositions and mechanical tensile properties vary widely in different ranges whereas the processing and compression property details remain the same for most of the row level data and can be grouped together. With this observation, the schema design chosen for intelligent data is the embedded data model, because of the nature of the possibility of data grouping. The

data storage is done using nested documents. By using this schema design, we can store 217 rows of data as 143 documents. Each document is stored as a key-value pair. The key of each document is the unique Id, and Value holds 39 fields; one of the fields is named "Metal Properties", which holds the embedded documents. These embedded documents may have multiple elements forming the sub-embedded documents and each element has 30 fields internally. If there are multiple data rows related to one single article, then the "Metal Properties" field holds multiple elements and each element in turn holds 30 fields. The embedded and sub-embedded documents also hold the data in the form of key-value pairs. Figure 2 shows the data model design for the Mg-Alloy data.

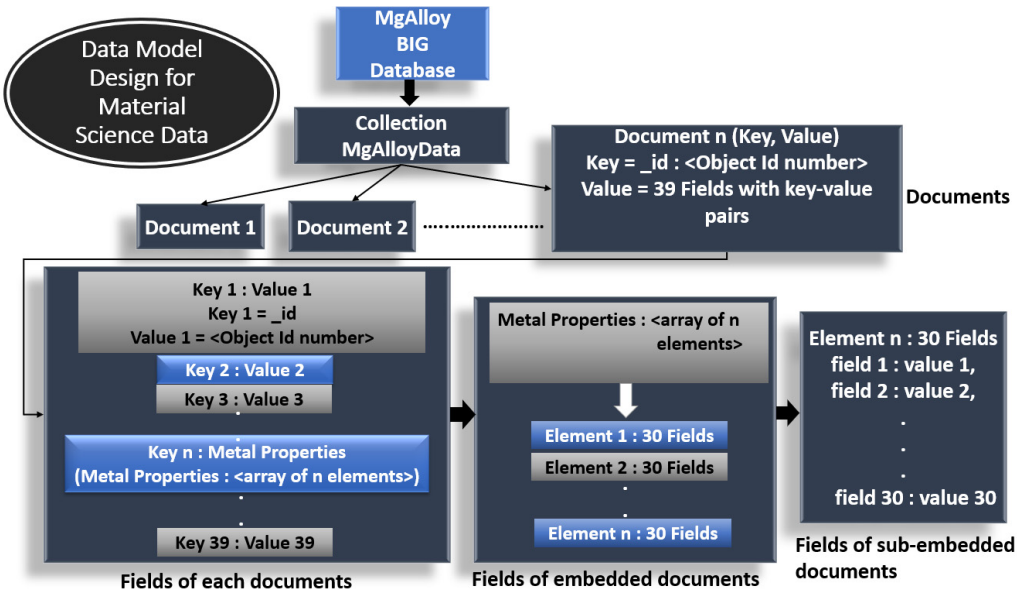

Figure 2. Data Model Design for Material Science data.

4.3. Data Preprocessing

The raw data is cleaned manually to remove the merged cells of rows and columns in a CSV file. The special characters are removed from the data and the handling of numerical missing data in Mg-alloy parameters is done using NANS (Not A Number), as the empty cells do not add any information to the actual Mg-alloy data because only few metals compositions are combined to form Mg-alloy and the rest of the metal composition percent is not present. If a scientific article has multiple rows of data, then information related to the author, title, nationality of the author, institute of the author, and publication source link are repeated in the first five columns of the CSV file during this phase. This redundancy is eliminated by grouping these columns combined with processing and corrosion properties in the embedded document while converting CSV file to JSON file format.

4.4. Data Conversion Model

The data in mongo DB can be stored in the form of a JSON object [13]. The semi-structured preprocessed data is converted from CSV to JSON file format and is stored as a nested document in the database. The details of scientific articles like, author, title, nationality of the author, institute of the author, and publication source link are grouped together with processing and corrosion properties of the Mg-alloy which belong to the main document. Each Mg-alloy has unique metal composition and mechanical properties; these two properties are grouped together to form embedded documents in the main document. If each article has multiple Mg-alloy metal combinations, then we have multiple embedded documents in the main document; embedded documents in turn may contain multiple sub-embedded documents. All the main documents combined form a collection. This

collection is a JSON file that is ready to be imported into MongoDB. We have 217 rows of data collected and transformed into 143 documents in the JSON file format.

4.5. Data Transfer Model

We used a standalone MongoDB cluster on the local machine by downloading and installing MongoDB on the local machine as per the documentation provided in the MongoDB official website [13]. We can start the server on the machine and establish the connection using mongo shell from the command prompt or using the Mongo compass GUI tool by providing the connection string. Once the connection is successful, we have created a database and named the collection. Browse the JSON file i.e., collection saved on your machine and add it to the database. Now the data is successfully imported onto the NOSQL database. We can add, update, or delete the documents using any Mongo GUI tool like Mongo Compass or Studio 3T GUI. We have used studio 3T GUI for any kind of database operations on MongoDB and the JSON file is imported into MongoDB. We are successfully able to import 143 documents onto the database.

5. Big Data Mining

There are 39 fields for each document which includes the Unique Id of the document. One of the fields being "Metal Properties" which forms the nested embedded document in the main document. This embedded document can hold n elements and each element holds 30 more fields which includes the information of metal compositions and their mechanical and corrosion properties. Our focus is to extract the mechanical tensile properties of the metals, as this property is the deciding factor for the ductility of the Mg-alloy. To retrieve these values from the database. Firstly, establish a connection through MongoClient function which can be imported from the PyMongo Library in the python script. Once the connection is successfully established, specify the database and collection name to access the data from the collection. Based on the field names, build a query, and apply projections to retrieve the required data. Below are the two sample queries that we have used to extract the mechanical tensile properties of the alloy. The result will be a JSON object.

5.1. Query the Document to Retrieve Tensile Properties of the Alloy

5.1.1. Query to Retrieve All Existing Tensile Values

The document in the MongoDB is queried by setting filter values to exist for all the three tensile property values and "OR" condition is applied among the tensile values. The projection is to include the tensile column values in the result set. In the projection, if we choose the option as include, then the column value is set to one in the query code. We have used Mongo Visual Query builder to build the query and to set the filters and projections for retrieving tensile properties. Pseudocode 1 shows the query code for retrieving all existing tensile values from the database.

Pseudocode 1. for the Query to retrieve all tensile values.

```
Use database; # Specify the database name.
# Specify the collection name in find query.
db.getCollection("Collection name").find(
{
#Apply OR condition for tensile properties.
"$or": [
{
      # Add filters
{#check if "Tensile UTS" exists}},
{#check if "Tensile YS" exists }},
{#check if "Tensile EL" exists }}
}
]
},
```

```
# Add projections
{
#Include projection for UTS
#Include projection for YS
#Include projection for EL
}
);
```

5.1.2. Query to Retrieve All Tensile Details Whose Tensile Values ≥ 0 or Not NAN's

The document in the MongoDB is queried by setting filter values that do not equal to NAN for all the three tensile property values and "AND" condition is applied among the tensile values. The projection is to include the tensile column values in the result set. In the projection, if we choose the option as include, then the column value is set to one; otherwise, if we choose to exclude the column, value is set to zero in the query code. Pseudocode 2 shows the query code to retrieve all numerical tensile values.

Pseudocode 2. for the Query to retrieve all numerical tensile values.

```
Use database; # Specify the database name.
# Specify the collection name in find query.
db.getCollection("Collection name").find(
{
#Apply AND condition for tensile properties.
"$and": [
{
# Add filters
{"Tensile UTS": { not equals: NAN}},
{"Tensile YS": { not equals: NAN}},
{"Tensile EL": not equals: NAN}}
}
]
},
# Add projections
{
#Include projection for UTS
#Include projection for YS
#Include projection for EL
}
);
```

5.1.3. Pseudocode Execution Flow

Figure 3 shows the steps of execution of the query codes in Pseudocode 1 and Pseudocode 2, and we can see that the collection scan is applied on the Database.CollectionName. The specified filters and projections are applied on query code of Pseudocode 1 and Pseudocode 2 to retrieve the tensile values, and the result is displayed.

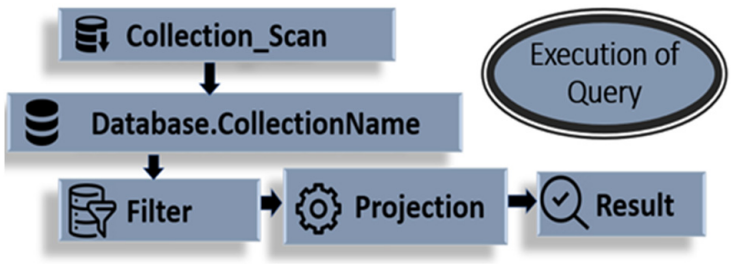

Figure 3. Execution flow of Pseudocode 1 and Pseudocode 2.

5.2. Convert JSON to CSV File Format and Prepare the Dataset to Feed the Machine Learning Model

The document is queried using the query code shown in Pseudocode 1 to retrieve all existing tensile values, and we can retrieve 143 documents. This result set is in the form of JSON object. This JSON object is parsed and written to a CSV file using python scripting which forms an input to the ML algorithm to perform classification of Mg-alloy. When JSON to CSV conversion is done we have observed 217 rows of tensile values which matches the count of original dataset. Figure 4 shows the sample of 20 rows of retrieved results.

	Id	Tensile Ys Mpa	Tensile UTS Mpa	Tensile E Percent
2	60b808e00a57f02374e2b135	277	NAN	11.5
3	60b808e00a57f02374e2b136	341.1	NAN	6
4	60b808e00a57f02374e2b137	161	NAN	27
5	60b808e00a57f02374e2b138	172	313	26.9
6	60b808e00a57f02374e2b138	179	320	25.3
7	60b808e00a57f02374e2b138	188	330	23.7
8	60b808e00a57f02374e2b138	206	332	13.4
9	60b808e00a57f02374e2b139	186	200	30.8
10	60b808e00a57f02374e2b139	284	292	25.5
11	60b808e00a57f02374e2b139	248	263	33.4
12	60b808e00a57f02374e2b139	277	292	20.2
13	60b808e00a57f02374e2b13a	NAN	NAN	NAN
14	60b808e00a57f02374e2b13a	NAN	NAN	NAN
15	60b808e00a57f02374e2b13a	NAN	NAN	NAN
16	60b808e00a57f02374e2b13a	NAN	NAN	NAN
17	60b808e00a57f02374e2b13a	NAN	NAN	NAN
18	60b808e00a57f02374e2b13a	NAN	NAN	NAN
19	60b808e00a57f02374e2b13a	NAN	NAN	NAN
20	60b808e00a57f02374e2b13a	NAN	NAN	NAN

Figure 4. Tensile properties retrieved from database using query from Pseudocode 1.

The query code of Pseudocode 2 is applied to exclude tensile properties with NAN values and the result has 108 documents with continuous numerical values for all three tensile properties. These 108 retrieved documents are in JSON format and are parsed to CSV to form 161 rows of data in CSV format. Figure 5 shows a sample of 20 rows of retrieved results.

	Id	Tensile Ys Mpa	Tensile UTS Mpa	Tensile E Percent
2	60b808e00a57f02374e2b138	172	313	26.9
3	60b808e00a57f02374e2b138	179	320	25.3
4	60b808e00a57f02374e2b138	188	330	23.7
5	60b808e00a57f02374e2b138	206	332	13.4
6	60b808e00a57f02374e2b139	186	200	30.8
7	60b808e00a57f02374e2b139	284	292	25.5
8	60b808e00a57f02374e2b139	248	263	33.4
9	60b808e00a57f02374e2b139	277	292	20.2
10	60b808e00a57f02374e2b13b	331	349	8.2
11	60b808e00a57f02374e2b13c	419	461	3.6
12	60b808e00a57f02374e2b13d	473	542	8
13	60b808e00a57f02374e2b13e	305.7	362.5	6.2
14	60b808e00a57f02374e2b13f	357.5	362.5	2.9
15	60b808e00a57f02374e2b140	400.6	443.3	2.4
16	60b808e00a57f02374e2b141	376.5	420.8	2.8
17	60b808e00a57f02374e2b142	446.7	480	2.4
18	60b808e00a57f02374e2b143	481.6	517.4	2
19	60b808e00a57f02374e2b144	177	250	15
20	60b808e00a57f02374e2b144	168	246	14

Figure 5. Numerical Tensile properties retrieved from database using query from Pseudocode 2.

6. Data Classification

The logistic regression algorithm is a popular binary classification algorithm [14]. In this model, the probabilities describing the possible outcomes are modeled using a logistic function known as sigmoid function. This function estimates the probability of an outcome to a value of 0 or 1, which helps to determine the class of output variable. It is an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1. Usually, zero is predicted as negative class and 1 as positive class.

Below is an example of the logistic equation. Input values (x) are combined linearly using weights or coefficient values to predict an output value (y). The output value being modeled is a binary value (0 or 1) rather than a numeric value which denotes the probability of the class.

$$Y = e^{\wedge}(B0 + B1 * x)/1 + e^{\wedge}(B0 + B1 * x)$$

where

$B0$: Bias or intercept term

$B1$: Coefficient of single input value (x)

In the Mg-alloy data, we chose the tensile properties YS, UTS as the independent input variables to the algorithm to predict the dependent output variable ductility. We have applied a logistic regression model to classify the Mg-alloy data. The classification result one is classified as ductile metal and zero as brittle metal. The classification accuracies are calculated, and spot check algorithm comparison is done for all three classification models; namely, Logistic regression, Support vector machine (SVM) and Random forest.

7. Results and Discussion of Data Classification

The dataset contains mechanical tensile properties YS, UTS and EL. We have introduced a dummy categorical variable called ductility based on the value of EL. Ductility holds value as one if EL value is greater than 15 and holds value as zero if EL value is less than 15. We chose Tensile YS, UTS as independent input variables and ductility as the output variable. The dataset is now divided into test and train samples. Out of 161 data rows, 80 percent of the dataset, i.e., 128 rows form the train data set and 20 percent of dataset i.e., 33 rows form the test dataset. The independent input variables in the train dataset are normalized using Standard Scaler function before fitting to the model. The 10-fold cross validation is applied on the 128 rows and observed the accuracies of 70.25%, 75% and 71.92% for Logistic Regression, SVM, Random Forest, respectively. Figure 6 shows the spot-check algorithm comparison of all the three models.

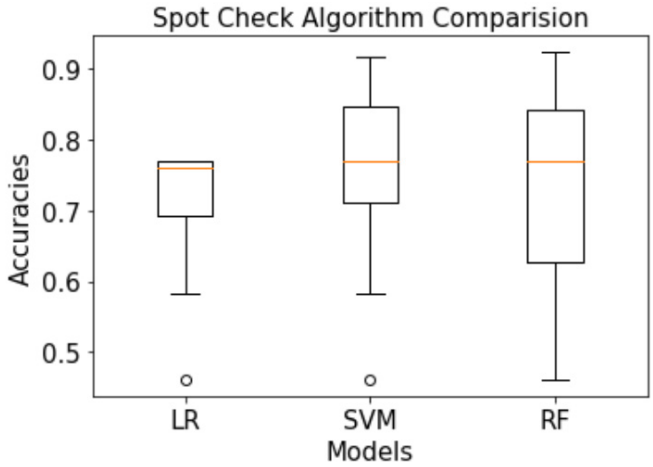

Figure 6. Spot Check Algorithm comparison for Logistic Regression, SVM and Random Forest models.

The predictions are made by fitting the transformed input values, output values of the train dataset (128 data rows) and tested on an independent dataset of 33 rows using the Logistic regression and SVM models, with observed accuracies of 72.72% and 78.78%,

respectively. Standardization did not improve the model performance and hence we used the original independent values with Random Forest, as the model is not sensitive to the magnitude of any two input variables. We have tested the model on an independent dataset of 33 rows and observed an accuracy of 87.87%. We noticed that Random Forest outperformed, with an increase of 15% and 9% against Logistic Regression and SVM models.

Table 3 shows the details of parameters that we have used for all three models.

Table 3. Model parameters for Logistic regression, SVM and Random Forest algorithms.

ML Models	Parameters
Logistic Regression	random_state = 0 solver = 'liblinear' penalty = l2
Support Vector Machine (SVM)-Model-1	kernel = 'rbf' random_state = 0 probability = True C = 1
Support Vector Machine (SVM)-Model-2	kernel = 'poly' degree = 8 random_state = 0 probability = True C = 1
Random Forest	n_estimators = 200 criterion = 'entropy' random_state = 0

For all three models, the random state parameter is set to zero to calculate the reproducible output across multiple function calls. For Logistic Regression, the parameter for solver is set to liblinear based on the smaller size of the dataset, l2 penalty is chosen to apply Ridge regression regularization method. For SVM, as the data is non-linearly separable, we have evaluated the model performance using two different kernels as listed in Table 3. Firstly, for SVM-Model-1 the kernel is set to the radial basis function (RBF), the regularization parameter C is set to 1 (the regularization strength is inversely proportional to C value) and the constructor option probability is set to True, to enable the class membership probability estimates (from the methods predict_proba and predict_log_proba). This probability variable is used to calculate the Receiver Operating Characteristic (ROC)—Area under curve (AUC) score to determine the performance of the classification model. Secondly, the kernel is changed to polynomial and the degree of polynomial is set to a numerical value 8 and the rest of the model parameters remain the same. When evaluated, both the models SVM-Model-1 and SVM-Model-2 equally performed, resulting in an accuracy of 78.78%. For Random Forest, we have set the model parameters n_estimators equal to 200 and criterion as "entropy". The parameter "n_estimators" is to specify the number of the trees in the forest and criterion: "entropy" for the information gain. Here we can also use Gini as criterion, but we have observed that model performed better with entropy for criterion.

Below is the comparison of performance metrics of Logistic regression, SVM and Random Forest models.

From Table 4, we can notice that Logistic regression has low performance when compared to the other models and Random Forest model has outperformed with a value of 0.88 for all the metrics- precision, recall, F1-Score, and accuracy.

Table 4. Performance Metrics of Mg-alloy dataset on the independent test data (33 rows) using the parameters from Table 1.

ML Models	Precision	Recall	F1-Score	Accuracy
Logistic Regression	0.71	0.73	0.73	0.73
SVM-Model-1	0.78	0.79	0.79	0.79
SVM-Model-2	0.80	0.79	0.77	0.79
Random Forest	0.88	0.88	0.88	0.88

For SVM, we have experimented with the polynomial kernel transformation of degree = 3 and observed the accuracy of 0.75, whereas with degree = 8 the accuracy is 0.78. However Radial-basis function (RBF) gave the similar accuracy equals to 0.78 when regularization parameter C value is 1.0. Based on this observation, we can say that polynomial kernel with degree = 3 underperformed when compared to RBF because the characteristics of the data distribution falls in the radial basis curve. Despite the lower accuracy, the non-linear polynomial kernel SVM-model with degree = 3 is preferred rather than degree = 8 as it is difficult to generalize the model with higher polynomial degree.

With these results, we can determine the best model for the Mg-alloy dataset, but before making the decision, we also calculated one of the important metrics Receiver Operating characteristics (ROC) and the Area under curve (AUC)-ROC. ROC curves are very helpful to understand the balance between true-positive rate and false positive rates. For calculating Receiver Operating Characteristic Curve (ROC), the probabilities are predicted for the independent variables of the test dataset (33 rows) and only positive probability is considered to calculate the ROC_AUC scores. The True positive and False positive rates are calculated which forms an input to plot the roc-curve. The Area under curve (AUC) for ROC is calculated for all the three models and tabulated in Table 5, and ROC curve plots are shown in Figure 7.

Table 5. ROC-AUC values for Logistic regression, SVM and Random Forest models.

ML Models	ROC-AUC Score
Logistic Regression	0.603
Support Vector Machine (SVM)-Model-1	0.777
Support Vector Machine (SVM)-Model-2	0.835
Random Forest	0.835

Table 5 shows the ROC-AUC scores for all three models. The ROC-AUC values range from 0 to 1. The models with ROC_AUC score close to 1 has better performance compared to others.

From Table 5, we observe that SVM-Model-2 and Random Forest have outperformed compared to the other two models. The AUC_ROC value for SVM-model-2 is high as equal as Random Forest but we do not prefer the SVM-Model-2, as this model is built using polynomial kernel with a higher degree = 8 which is difficult to generalize the model. Despite that, the SVM model with polynomial kernel of degree = 3 is preferred, achieving an AUC_ROC value of 0.818. Hence, we can consider Random Forest and SVM with polynomial kernel of degree = 3 as the best classification models for the Mg-alloy dataset.

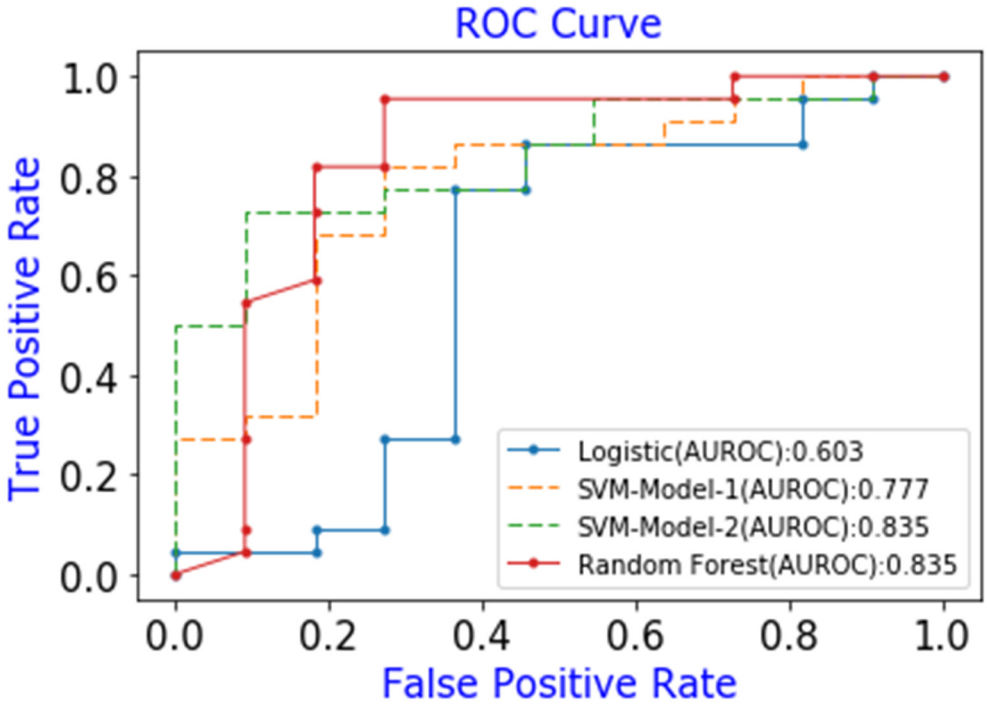

Figure 7. ROC curves for Logistic Regression, SVM and Random Forest models.

In addition to the classification models, we applied regression analysis for the Mg-alloy dataset using the multivariate polynomial regression, because we have two varying independent variables. We chose YS and UTS as the independent variables to predict the values of dependent variable EL. The independent variables were transformed using the polynomial function of degree = 3 and the transformed values of YS and UTS, and original EL values are then split into the into test and train datasets. Out of 161 rows, 80% of dataset forms train data (128 rows) and the remaining 20% of the dataset (33 rows) as test data. The train data is now fit to the polynomial regression model to predict the EL values. The regression model evaluation is done by calculating the metrics, Root Mean square error (RMSE) and Pearson Correlation Coefficient (R-squared value, r^2) to measure the amount of variation of predicted values from actual values. Here, we have observed an RMSE value equal to 5.57 and R-squared value equal to 0.60. The R-squared value for ideal regression model must equal 1, but here the observed score is 0.60. This regression scores coincides similarly to the limited classification accuracy performance (less than 80% cross-validation accuracy).

We believe that the ductility model could be dependent further on metal compositions, processing parameters/methods to yield higher Pearson correlation coefficient and classification accuracies. Alternatively, we anticipate further improving the polynomial regression model using data transformation techniques.

8. Conclusions

In this study, we developed the first ever big data storage for the Mg-alloy data, which is highly essential to develop an intelligent database, automate the machine learning models for determination of strong Mg-alloy materials, and further perform material science analytics. We highly recommend that the MONGODB big data tool has outperformed for the storage of the semi-structured Mg-alloy big data. We observed that the big data mining operations, of insertion and retrieval consumed minimal time in the MONGODB big database. We also experimented with Apache Hive for Mg-alloy data storage and observed that Hive did not perform well when compared to MONGODB because of the nature of the connectivity of the data elements particularly in Mg-alloy database [50]. Also, the classification results of the ductility model show that we achieved significant accuracies

of up to 75% among all three machine learning classification models. The regression results also indicated limited R-squared value of 0.60 similar to the classification accuracies. There is scope to improve the classification accuracies and regression results, by applying data transformation techniques.

We can extend this study by (i) expanding the input variables to include the metal composition values, processing methods/parameters, and tensile properties to determine the ductility of the metal. However, these variables are a highly sparse dataset depending on the nature of the choice of metal compositions in different iterations. We would like to address this problem of a sparse dataset by exploring the sub-sampling methods and singular-value decomposition (SVD) of ML techniques. In future work, (ii) we would also like to develop the inverse model of predicting metal composition, processing methods, and their parameters by considering tensile properties and ductility of the metal as independent input variables.

Author Contributions: Conceptualization, Z.X. and B.G.; methodology, B.G., K.R. and S.C.; software, S.C.; validation, B.G.; investigation, B.G. and S.C.; resources, J.S., Z.X. and S.C.; data curation, Z.X. and S.C.; writing—original draft preparation, S.C.; writing—review and editing, B.G., Z.X. and S.C.; visualization, B.G. and S.C.; supervision, B.G., Z.X., J.S. and K.R.; project administration, Z.X. and J.S.; funding acquisition, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by NSF Engineering Research Center for Revolutionizing Metallic Biomaterials. NSF EEC (ERC 0812348),Army Research Laboratory PRI Program (W911NF-12-2-0022), and NSF CMMI 2026313.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

Mg-alloy	Magnesium alloy, Intelligent Material science data
SVM	Support vector machine
ML	Machine Learning
NoSQL	non-SQL, non-Structured Query Language or non-relational
RDBMS	Relational database management systems
SQL	Structured Query Language or relational
EHR	Electronic health records
CSV	Comma separated value
JSON	Java script object notation
GUI	Graphical user interface
NAN	Not A number
YS	Tensile yield strength
UTS	Ultimate tensile strength
EL	Elongation-at-fracture
RMSE	Root mean square error
R^2	R-squared, Pearson Correlation Coefficient
SVD	Singular value decomposition

References

1. Xu, T.; Yang, Y.; Peng, X.; Song, J.; Pan, F. Overview of advancement and development trend on magnesium alloy. *J. Magnes. Alloy.* **2019**, *7*, 536–544. [CrossRef]
2. Mathaudhu, S.N.; Nyberg, E.A. Magnesium alloys in U.S. military applications: Past, current and future solutions. In *Essential Readings in Magnesium Technology*; Mathaudhu, S.N., Luo, A.A., Neelameggham, N.R., Nyberg, E.A., Sillekens, W.H., Eds.; Springer: Cham, Switzerland, 2016. [CrossRef]
3. Jones, T.L.; Kondoh, K. Ballistic analysis of new military grade magnesium alloys for armor applications. In *Magnesium Technology 2011*; Sillekens, W.H., Agnew, S.R., Neelameggham, N.R., Mathaudhu, S.N., Eds.; Springer: Cham, Switzerland, 2011. [CrossRef]
4. Luo, A.A. Magnesium casting technology for structural applications. *J. Magnes. Alloy.* **2013**, *1*, 2–22. [CrossRef]

5. Li, X.; Qi, W.; Zheng, K.; Zhou, N. Enhanced strength and ductility of Mg–Gd–Y–Zr alloys by secondary extrusion. *J. Magnes. Alloy.* **2013**, *1*, 54–63. [CrossRef]
6. Sun, H.-F.; Li, C.-J.; Xie, Y.; Fang, W.-B. Microstructures and mechanical properties of pure magnesium bars by high ratio extrusion and its subsequent annealing treatment. *Trans. Nonferrous Met. Soc. China* **2012**, *22* (Suppl. 2), s445–s449. [CrossRef]
7. Cheng, R.; Li, M.; Du, S.; Pan, H.; Liu, Y.; Gao, M.; Zhang, X.; Huang, Q.; Yang, C.; Ma, L.; et al. Effects of single-pass large-strain rolling on microstructure and mechanical properties of Mg-Al-Ca alloy sheet. *Mater. Sci. Eng. A* **2020**, *786*, 139332. [CrossRef]
8. Zanchetta, B.D.; da Silva, V.K.; Sordi, V.L.; Rubert, J.B.; Kliauga, A.M. Effect of asymmetric rolling under high friction co-efficient on recrystallization texture and plastic anisotropy of AA1050 alloy. *Trans. Nonferrous Met. Soc. China* **2019**, *29*, 2262–2272. [CrossRef]
9. Zhang, H.; Xu, Z.; Yarmolenko, S.; Kecskes, L.; Sankar, J. Evolution of Microstructure and Mechanical Properties of Mg-6Al Alloy Processed by Differential Speed Rolling upon Post-Annealing Treatment. *Metals* **2021**, *11*, 926. [CrossRef]
10. Biswas, S.; Suwas, S. Evolution of sub-micron grain size and weak texture in magnesium alloy Mg–3Al–0.4 Mn by a modified multi-axial forging process. *Scr. Mater.* **2012**, *66*, 89–92. [CrossRef]
11. Hong, M.; Wu, D.; Chen, R.; Du, X. Ductility enhancement of EW75 alloy by multi-directional forging. *J. Magnes. Alloy.* **2014**, *2*, 317–324. [CrossRef]
12. Prasetio, D.; Harlili, D.; Sc, M. Predicting football Match Results with Logistic Regression. In Proceedings of the 2016 International Conference on Advanced Informatics: Concepts, Theory And Application (ICAICTA), Penang, Malaysia, 16–19 August 2016.
13. MongoDB Documentation Team. Structure Your Data for MongoDB. Available online: https://docs.mongodb.com/guides/server/introduction/ (accessed on 1 October 2020).
14. Ezukwoke, K.I.; Zareian, S.J. *Logistic Regression and Kernel Logistic Regression—A Comparative Study of Logistic Regression and Kernel Logistic Regression for Binary Classification*; University Jean Monnet: Saint-Etienne, France, 2019. [CrossRef]
15. Madhava, V.; Sreekanth, R.; Nanduri, S. Big Data Electronic Health Records Data Management and Analysis on Cloud with MongoDB: A NoSQL Database. *Int. J. Adv. Eng. Glob. Technol.* **2015**, *3*, 943–949.
16. Patil, M.; Hanni, A.; Tejeshwar, C.H.; Patil, P. A qualitative analysis of the performance of MongoDB vs. MySQL database based on insertion and retriewal operations using a web/android application to explore load balancing—Sharding in MongoDB and its advantages. In Proceedings of the 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 10–11 February 2017. [CrossRef]
17. Li, Z. NoSQL Databases. In *The Geographic Information Science & Technology Body of Knowledge*, 2nd Quarter 2018 ed.; Wilson, J.P., Ed.; 2018; Available online: https://gistbok.ucgis.org/bok-topics/nosql-databases. (accessed on 3 September 2021). [CrossRef]
18. Kowsari, K.; Meimandi, J.K.; Heidarysafa, M.; Mendu, S.; Barnes, L.; Brown, D. Text Classification Algorithms: A Survey. *Information* **2019**, *10*, 150. [CrossRef]
19. Yan, R.; Xia, Z.; Xie, Y.; Wang, X.; Song, Z. Research on Sentiment Classification Algorithms on Online Review. *Complexity* **2020**, *2020*, 5093620. [CrossRef]
20. Gokaraju, B.; Durbha, S.S.; King, R.L.; Younan, N.H. A Machine Learning Based Spatio-Temporal Data Mining Approach for Detection of Harmful Algal Blooms in the Gulf of Mexico. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2011**, *4*, 710–720. [CrossRef]
21. Palacharla, P.K.; Durbha, S.S.; King, R.L.; Gokaraju, B.; Lawrence, G.W. A hyperspectral reflectance data based model inversion methodology to detect reniform nematodes in cotton. In Proceedings of the 2011 6th International Workshop on the Analysis of Multi-temporal Remote Sensing Images (Multi-Temp), Trento, Italy, 12–14 July 2011; pp. 249–252. [CrossRef]
22. Gokaraju, B.; Nóbrega, R.A.A.; Doss, D.A.; Turlapaty, A.C.; Tesiero, R.C. Data fusion of multi-source satellite data sets for cost-effective disaster management studies. *SoutheastCon* **2017**, *2017*, 1–5. [CrossRef]
23. Gokaraju, B.; Durbha, S.S.; King, R.L.; Younan, N.H. Sensor web and data mining approaches for Harmful algal bloom detection and monitoring in the Gulf of Mexico region. In Proceedings of the 2009 IEEE International Geoscience and Remote Sensing Symposium, Cape Town, South Africa, 12–17 July 2009; pp. III-789–III-792. [CrossRef]
24. Gokaraju, B.; Agrawal, R.; Doss, D.A.; Bhattacharya, S. Identification of Spatio- Temporal Patterns in Cyber Security for Detecting the Signature Identity of Hacker. *SoutheastCon* **2018**, *2018*, 1–5. [CrossRef]
25. Chen, Y.; Xu, Z.; Smith, C.; Sankar, J. Recent advances on the development of magnesium alloys for biodegradable implants. *Acta Biomater.* **2014**, *10*, 4561–4573. [CrossRef]
26. Xu, L.; Yu, G.; Zhang, E.; Pan, F.; Yang, K. In vivo corrosion behavior of Mg-Mn-Zn alloy for bone implant application. *J. Biomed. Mater. Res. Part A* **2007**, *83A*, 703–711. [CrossRef]
27. Witte, F.; Kaese, V.; Haferkamp, H.; Switzer, E.; Meyer-Lindenberg, A.; Wirth, C.; Windhagen, H. In vivo corrosion of four magnesium alloys and the associated bone response. *Biomaterials* **2005**, *26*, 3557–3563. [CrossRef] [PubMed]
28. Ullmann, B.; Reifenrath, J.; Dziuba, D.; Seitz, J.-M.; Bormann, D.; Meyer-Lindenberg, A. In vivo degradation behavior of the magnesium alloy LANd442 in rabbit tibiae. *Materials* **2011**, *4*, 2197–2218. [CrossRef] [PubMed]
29. Easton, M.; Beer, A.G.; Barnett, M.R.; Davies, C.; Dunlop, G.; Durandet, Y.; Blacket, S.; Hilditch, T.; Beggs, P.D. Magnesium alloy applications in automotive structures. *JOM* **2008**, *60*, 57. [CrossRef]
30. Luo, A.A. Recent magnesium alloy development for automotive powertrain applications. *Mater. Sci. Forum* **2003**, *419–422*, 57–66. [CrossRef]

31. Kulekci, M.K. Magnesium and its alloys applications in automotive industry. *Int. J. Adv. Manuf. Technol.* **2008**, *39*, 851–865. [CrossRef]
32. Staiger, M.P.; Pietak, A.M.; Huadmai, J.; Dias, G. Magnesium and Its Alloys as Orthopedic Biomaterials: A Review. *Biomaterials* **2006**, *27*, 1728–1734. [CrossRef]
33. Kim, W.J.; Lee, Y.G. High-strength Mg–Al–Ca alloy with ultrafine grain size sensitive to strain rate. *Mater. Sci. Eng. A* **2011**, *528*, 2062–2066. [CrossRef]
34. Bae, S.W.; Kim, S.-H.; Lee, J.U.; Jo, W.-K.; Hong, W.-H.; Kim, W.; Park, S.H. Improvement of mechanical properties and reduction of yield asymmetry of extruded Mg-Al-Zn alloy through Sn addition. *Alloy. Compd.* **2018**, *766*, 748–758. [CrossRef]
35. Peng, P.; He, X.; She, J.; Tang, A.; Rashad, M.; Zhou, S.; Zhang, G.; Mi, X.; Pan, F. Novel low-cost magnesium alloys with high yield strength and plasticity. *Mater. Sci. Eng. A* **2019**, *766*, 138332. [CrossRef]
36. Kozlov, A.; Ohno, M.; Arroyave, R.; Liu, Z.-K.; Schmid-Fetzer, R. Phase equilibria, thermodynamics and solidification microstructures of Mg–Sn–Ca alloys, Part 1: Experimental investigation and thermodynamic modeling of the ternary Mg–Sn–Ca system. *Intermetallics* **2008**, *16*, 299–315. [CrossRef]
37. Abramova, V.; Bernardino, J. NoSQL databases: MongoDB vs cassandra. In Proceedings of the International C* Conference on Computer Science & Software Engineering, Porto, Portugal, 10–12 July 2013; Available online: https://dl.acm.org/doi/proceedings/10.1145/2494444. (accessed on 5 September 2021).
38. Sahatqija, K.; Ajdari, J.; Zenuni, X.; Raufi, B.; Ismaili, F. Comparison between relational and NOSQL databases. In Proceedings of the 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 21–25 May 2018. [CrossRef]
39. Silberstein, A.; Chen, J.; Lomax, D.; McMillan, B.; Mortazavi, M.; Narayan, P.; Ramakrishnan, R.; Sears, R. PNUTS in Flight: Web-Scale Data Serving at Yahoo. *IEEE Internet Comput.* **2012**, *16*, 13. [CrossRef]
40. No-SQL Databases. Available online: https://hostingdata.co.uk/nosql-database/ (accessed on 10 October 2020).
41. Han, J.; Haihong, E.; Le, G.; Du, J. Survey on NoSQL Database. In Proceedings of the 2011 6th international conference on pervasive computing and applications, Port Elizabeth, South Africa, 26–28 October 2011. [CrossRef]
42. Xu, X.; Wang, L.; Zhu, G.; Zeng, X. Predicting Tensile Properties of AZ31 Magnesium Alloys by Machine Learning. *JOM* **2020**, *72*, 3935–3942. [CrossRef]
43. Wu, L.; Li, M. Applying the CG-logistic Regression Method to Predict the Customer Churn Problem. In Proceedings of the 2018 5th International Conference on Industrial Economics System and Industrial Security Engineering (IEIS), Toronto, ON, Canada, 3–6 August 2018.
44. Chakraborty, D.; Sur, U.; Banerjee, P.K. Random Forest Based Fault Classification Technique for Active Power System Networks. In Proceedings of the 2019 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE), Bangalore, India, 15–16 November 2019.
45. Kwon, H. Friend-Guard Textfooler Attack on Text Classification System. *IEEE Access* **2021**, *4*, 99. [CrossRef]
46. Iyyer, M.; Wieting, J.; Gimpel, K.; Zettlemoyer, L. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv* **2018**, arXiv:1804.06059.
47. He, W.; Wei, J.; Chen, X.; Carlini, N.; Song, D. Adversarial example defense: Ensembles of weak defenses are not strong. In Proceedings of the 11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17), WOOT'17: Proceedings of the 11th USENIX Conference on Offensive Technologies, Vancouver, BC, Canada, 14–15 August 2017.
48. Tramer, F.; Carlini, N.; Brendel, W.; Madry, A. On adaptive attacks to adversarial example defenses. *arXiv* **2020**, arXiv:2002.08347.
49. Kwon, H.; Yoon, H.; Park, K.W. Multi-targeted backdoor: Identifying backdoor attack for multiple deep neural networks. *IEICE Trans. Inf. Syst.* **2020**, *103*, 883–887. [CrossRef]
50. Dhanya Nary Biju Yojna Arora, Department of Computer Science &Engineering Department of Computer Science & Engineering Amity University, Haryana, India. "Twitter Data Analysis using Hadoop", Vol-4 Issue-5 2018, IJARIIE-ISSN(O)-2395-4396. Available online: http://ijariie.com/AdminUploadPdf/Twitter_Data_Analysis_using_Hadoop_ijariie9093.pdf. (accessed on 3 September 2021).