

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Aplikace pro analýzu testovacích dat vakuových vývěv



2023

Vedoucí práce:
Mgr. Radek Janošík, Ph.D.

Jakub Pernický

Studijní program: Informační technologie,
prezenční forma

Bibliografické údaje

Autor: Jakub Pernický
Název práce: Aplikace pro analýzu testovacích dat vakuových vývěv
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2023
Studijní program: Informační technologie, prezenční forma
Vedoucí práce: Mgr. Radek Janoščík, Ph.D.
Počet stran: 36
Přílohy: elektronická data v úložišti katedry informatiky
Jazyk práce: český

Bibliographic info

Author: Jakub Pernický
Title: Application for analysing testing data of vacuum pumps
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2023
Study program: Information Technologies, full-time form
Supervisor: Mgr. Radek Janoščík, Ph.D.
Page count: 36
Supplements: electronic data in the storage of department of computer science
Thesis language: Czech

Anotace

System, který je schopný zpracovat desítky GB dat z testů vakuových vývěv během nízkých desítek minut. Zrychluje proces analýzy těchto dat uživatelem díky přívětivému procesu zpracování a detailní reprezentaci zpracovaných dat. Uživatel zadáním hodnot definuje anomální stavy běhů zaznamenaných ve vstupních datech a systém vyhledá a označí tyto anomálie. Uživatel má poté k dispozici grafy obsahující anomálie a soubory se zjednodušenými daty.

Synopsis

A system that is capable of processing tens of GB of vacuum pump test data in the low tens of minutes. It speeds up the process of analyzing this data by the user thanks to its user-friendly processing and detailed representation of the processed data. By entering values, the user defines anomalous run conditions recorded in the input data and the system locates and flags these anomalies. The user is then provided with graphs containing the anomalies and simplified data files.

Klíčová slova: vakuové vývěvy; testovací data; csv; analýza; anomálie

Keywords: vacuum pumps; testing data; csv; analysis, anomaly

Chtěl bych poděkovat panu Mgr. Radkovi Janoščíkovi, Ph.D., za trpělivost a důvěru v mou schopnost zpracovat obsah této práce. Také chci poděkovat firmě Edwards za umožnění zpracování tohoto tématu a Adamovi Novákovi za jeho pomoc a aktivní komunikaci při upřesňování detailů práce.

Odevzdáním tohoto textu jeho autor/ka místopřísežně prohlašuje, že celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

Obsah

1 Úvod	7
2 Řešená problematika	8
2.1 Vakuové vývěvy	8
2.2 Existující řešení	10
3 Uživatelský přístup	11
3.1 Vstupy	11
3.2 Nastavení požadavků	13
3.3 Výstupy	15
4 Programové řešení	17
4.1 Použité jazyky a technologie	17
4.1.1 C#	17
4.1.2 WPF	17
4.1.3 .NET	17
4.1.4 XAML	17
4.1.5 Použité knihovny	18
4.2 Architektura aplikace	18
4.3 Ošetření vstupů	21
4.4 Reprezentace dat	25
4.5 Zpracování dat	25
4.6 Reprezentace výstupů	26
5 Poznatky	28
5.1 Kvalita řešení	28
5.2 Budoucí vývoj	31
Závěr	32
Conclusions	33
A Obsah elektronických dat	34
Literatura	35

Seznam obrázků

1	Typy vakuových vývěv od <i>Edwards</i> . Zdroj: [9]	9
2	Turbomolekulární vývěva od <i>Edwards</i> . Zdroj: [10]	9
3	Vstup ve formátu <i>RAW</i> .	12
4	Vstup ve formátu <i>Results</i> .	12
5	Vyzvání ke zvolení vstupu.	12
6	Výběr vývěv a jejich vstupů.	13
7	Vyplnění hodnot a odchylek.	14
8	Výběr grafů k zobrazení.	15
9	Ukázka grafu.	15
10	Ukázka <i>RAW</i> výstupu.	16
11	Ukázka výstupu s anomálním cyklem.	16
12	Složení aplikace.	18
13	Diagram chodu aplikace.	20
14	Chyba při výběru vstupu.	21
15	Nevalidní vstupy.	22
16	Validní vstupy.	22
17	Zobrazení nevalidních a validních vstupů.	22
18	Při nezvolení vstupu pro zpracování.	23
19	Při nezadání hodnot a odchylek.	23
20	Při zadání pouze hodnoty nebo odchylky.	24
21	Nepropojené odchylky.	24
22	Propojené odchylky.	24
23	Graf s více anomálními cykly.	27
24	Anomálie ve stavu běhu.	27
25	Ukázka query pro anomálie v <i>Power</i> sloupci.	28
26	Výstup z query.	29
27	Vstup na test rychlosti.	29
28	Ukázka anomálií ve stavu běhu.	30

1 Úvod

Vývoj technologií a postupů ve firmách je stále rychlejší a s nimi také potřeba efektivnějšího testování a zpracovávání informací. V průběhu studia na střední a vysoké škole jsem měl možnost několik let pracovat ve firmě *Edwards Vacuum* v Lutíně (dále jen *Edwards*) a na vlastní kůži si tento vývoj zažít. Při testování vakuových vývěv je potřeba zaznamenávat informace o jejich běhu neboli o spotřebě, teplotě jednotlivých částí, rychlosti otáček a vytvořeném vakuu. Zavedeným standardem bylo tyto data kontrolovat pomocí *Excelu* a v případě krátkých testů na samostatných vývěvách byl tento postup udržitelný. Problém nastal při škálování testů. Při zavádění nových produktů do výroby nebo stížnosti zákazníka je potřeba dostatečně do hloubky tyto produkty otestovat. Tyto testy mohou běžet současně na desítkách vývěv po dobu několika měsíců. S tím ale nastává problém efektivní kontroly těchto testovacích dat, protože jejich manuální kontrola pomocí *Excelu* je pro tento objem dat pomalá a složitá.

Součástí mé práce pro *Edwards* bylo zpracovávání právě těchto testovacích dat. Velmi rychle jsem si uvědomil, že zavedené postupy nejsou dostačující a hledal jsem řešení na tento rostoucí problém. Po pokusech o zavedení nových způsobů analýzy testovacích dat jsem narazil na problémy s optimalizací a uživatelskou přívětivostí. Došel jsem tedy k názoru, že nejlepším řešením bude vytvoření aplikace stavěné na míru, podle požadavků firmy.

2 Řešená problematika

Má práce se zaměřuje na zpracovávání dat z vakuových vývěv. Vytváření vakua je neustále se vyvíjející odvětví, které je nedílnou součástí moderních technologií. Tato kapitola se zaměřuje na zasazení mé práce do kontextu tohoto odvětví.

2.1 Vakuové vývěvy

Vakuové vývěvy jsou využívány v mnoha odvětvích. Mezi ně spadá výroba analytických přístrojů, chemický a potravinářský průmysl, technologie jako urychlovače částic, lasery, výroba energie, polovodiče nebo displeje.[1, 2, 3, 4, 5, 6]

Existují různé typy vakuových vývěv, které se zaměřují na specifické oblasti využití (obrázek 1). V mé praxi jsem se nejčastěji setkával s turbomolekulárními vývěvami (obrázek 2).

Turbomolekulární vývěvy jsou vícestupňové turbíny, kde se pomocí rotujících lopatek vytlačí molekuly plynu a tím vytvářejí vakuum.[7] Tyto lopatky mohou dosahovat rychlostí až 90 000 otáček za minutu a vytvářejí tlak menší než $5 \cdot 10^{-10}$ mbar. [8] V případě mechanického problému v těchto vysokých otáčkách může dojít k zadrhnutí a nefunkčnosti vývěvy, a tedy celého systému.

Vakuové vývěvy pracují s velmi malými rozměrovými tolerancemi, které zajišťují jejich bezproblémový chod. V případě i malých nepřesností mohou nastat katastrofické následky.

Při vývoji nových verzí vývěv a odstraňování problémů u existujících verzí je potřebné důkladné testování jejich běhu. Vývěvy běží v různých konfiguracích a dobách, zaznamenávají se hodnoty o jejich běhu a poté se analyzují. V současné době se tvoří záznamy z testování v intervalech 1 nebo 5 vteřin a jsou na výběr 2 módy záznamů. Jeden uchovává větší množství dat a druhý pouze zjednodušené hodnoty.

Vývěvy při běhu procházejí několika stavy. Ze *zastavených* (stopped) přecházejí do *zrychlujících se* (accelerating), po dosažení požadované rychlosti otáček přecházejí do stavu *běžící* (running) a v něm jsou do dokončení nastaveného typu běhu nebo do manuálního vypnutí. Poté přechází do stavu *zpomalujících se* (decelerating) a při nulových otáčkách do stavu *zastavené* (stopped). Sledováním chování hodnot ve stavu běžící je možné vypořizovat nežádoucí chování, které by v budoucnu mohlo vytvořit problémy v kvalitě tvořeného vakua nebo přímo způsobit nefunkčnost vývěvy.



Obrázek 1: Typy vakuových vývěv od *Edwards*. Zdroj: [9]



Obrázek 2: Turbomolekulární vývěva od *Edwards*. Zdroj: [10]

2.2 Existující řešení

Pro kontrolu testovacích dat bylo využíváno manuální vkládání souborů ve formátu *Comma-separated values* (dále jen *csv*) do *Excelu*, vytvoření grafů a vizuální kontrola jejich běhu. Tato metoda byla udržitelná v případě jednotek MB testovacích dat nebo na malém počtu testovaných vývěv. V této metodě nebylo ošetřeno přesné nalezení nechtěného chování, muselo se pouze spoléhat na znalosti uživatele, který data kontroloval. Při zavedení nových testovacích metod se zvýšil objem testovaných vývěv na desítky jednotek najednou a prodloužila se doba jejich testování. To způsobilo nashromáždění testovacích dat ve velkém množství v řádu desítek GB. Už zpracovávání jednotek MB zabralo nezkušenému uživateli několik hodin práce a při větším množství vývěv se mohlo jednat o jednotky dnů aktivní práce. Bylo tedy potřeba změnit přístup ke kontrole dat.

Jako řešení částečné automatizace kontroly a označení nechtěného chování (dále jen *anomálií*) bylo využito *PowerQuery*. *PowerQuery* je doplněk pro *Excel*, který umožňuje importovat nebo připojit a upravovat externí data. To umožnilo automaticky rozdělit data ze souborů podle jednotlivých vývěv, vytvořit průměry a mediány pro požadované sloupce, maxima a minima a označit soubory obsahující anomálie. Vzhledem k optimalizaci *PowerQuery* stále docházelo k problémům u objemu testovacích dat větším než nízké stovky MB, a při jejich zpracovávání přetékala operační paměť. Bylo tedy zapotřebí vstupní data rozdělit do menších bloků a ručně aplikovat vytvořené query zvláště pro každý blok. Toto řešení urychlilo kontrolu testovaných dat z dnů manuálního zpracovávání na hodiny, ale nebylo uživatelsky přívětivé. Uživatel také neměl jednoduchou a jasnou možnost měnit rozmezí pro označení anomálií. Také bylo potřeba používat jiné postupy při zpracovávání různých formátů vstupů. Tyto problémy vytvořily poptávku po efektivnějším způsobu zpracovávání testovacích dat z důvodu neustálého zvětšování objemu testovaných vývěv a doby jejich testování. Při výzkumu existujícího systému, který by splňoval požadavky pro analýzu těchto dat, jsem nenarazil na nic uspokojivého. Existují aplikace, které umožňují prohledávání *csv* souborů o několika GB, ale neposkytovaly možnosti zobrazení grafů nalezených hodnot a obecně nebyly velmi uživatelsky přívětivé. Jako řešení jsem zvolil vývoj vlastní aplikace na míru, která uživatele provede nastavením požadavků, vyhledá a označí anomálie a vytvoří upravené výstupy pro pozdější použití.

Hlavním očekáváním pro mé řešení bylo umožnění zpracování několika GB vstupů bez potřeby aktivní obsluhy. Vzhledem k neschopnosti předchozího řešení zpracovat takové množství vstupů najednou, byla potřeba aktivně spouštět zpracování nové dávky vstupních souborů po dokončení zpracování předchozí dávky. Mé řešení, po nastavení počátečních parametrů, zpracovává data bez nutnosti obsluhy. Uživatel také musel v předchozím řešení ručně filtrovat, které vstupy se budou zpracovávat, jejich rozdělením do jednotlivých složek.

Mé řešení má umožnit uživateli vybrat vstup s testovacími soubory a určit, jaké vývěvy, jejichž soubory jsou na vstupu, chce uživatel zkontrolovat. Poté může nastavit, které sloupce ze vstupních souborů se budou kontrolovat a určit rozmezí hodnot, mimo které bude běh vývěvy považován jako anomální. Po zpracování dat má uživatel možnost vidět vývěvy, které prošly kontrolou bez nalezení anomálie a vypočítané průměry, mediány a maxima ze zvolených sloupců, pro rychlou kontrolu běhu jsou vygenerovány ve výstupním souboru. Vývěvy s nalezenou anomálií jsou označeny jako anomální. Anomální vývěvy mají kromě stejného výstupu jako vývěvy bez anomálií navíc také soubor s daty anomálních cyklů pro pozdější zpracování. Uživatel má možnost si u anomálních vývěv zobrazit grafy pro jednotlivé soubory s nalezenými anomáliemi.

3 Uživatelský přístup

V této kapitole se zaměřím na pohled uživatele na mou aplikaci. Co přesně je považováno za vstupy, jakým způsobem uživatel definuje anomálie a co může očekávat na výstupu po zpracování zadaných vstupů.

3.1 Vstupy

Vstupy aplikace jsou csv soubory ve dvou různých formátech, *RAW* a *Results*. Nejčastěji jsou používány *RAW* záznamy, které obsahují čas záznamu, stav, rychlost otáček, spotřebu, teplotu motoru, rotoru a kontroléru, referenci rychlosti, vakuum a motohodiny (obrázek 3). Druhý formát záznamů je *Results*, který obsahuje čas rozběhu, hodnotu rozběhu, čas sjezdu, hodnotu sjezdu, čas záznamu teploty a hodnotu teploty (obrázek 4). Velikosti souborů *Results* se většinou pohybují v jednotkách kB, zatímco *RAW* soubory mohou nabývat desítek MB. V názvu souboru je zaznamenáno sériové číslo vývěvy a čas uložení záznamu. Uživatel požaduje vyhledání anomálií v obou typech vstupních souborů, sjednocení všech vstupů typu *Results*, výpočet průměrů, mediánů a nalezení maxim pro data ze vstupů typu *RAW*. V případě, že chce uživatel pouze zobrazit grafy již zpracovaných dat, vstupem je soubor se zpracovanými daty ve formátu csv nebo txt s názvem buď *AnomalyData_RAWSeriovéČíslo* nebo *AnomalyData_ResultsSeriovéČíslo*.

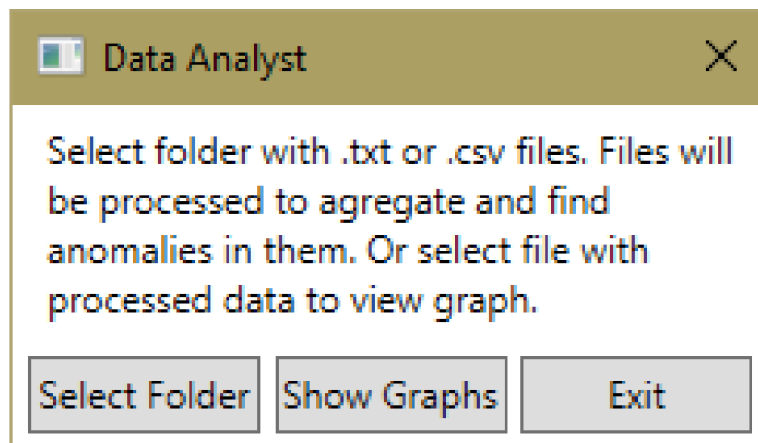
Po spuštění aplikace je uživatel vyzván ke zvolení složky se vstupními soubory nebo k zobrazení grafů z již vygenerovaných výstupů (obrázek 5). V případě nezvolení vstupní složky nebo souboru se zpracovanými daty je upozorněn a může aplikaci ukončit nebo zvolit nový vstup.

```
210464721_211001_14.11_RAW - Notepad
File Edit Format View Help
Date_Time;PUTstat;Speed;Power;MTemp;RTemp;CTemp;SpeedRef;Vacuum;MotoHours
10/1/2021 2:11:19 PM;Accelerating;647.5;1.2;49.0;60.6;40.5;1500.0;3.350E-1;205
10/1/2021 2:11:20 PM;Accelerating;653.0;29.0;48.9;60.6;40.5;1500.0;3.350E-1;205
10/1/2021 2:11:21 PM;Accelerating;661.9;47.0;48.9;60.6;40.5;1500.0;3.350E-1;205
10/1/2021 2:11:22 PM;Accelerating;673.5;62.6;48.9;60.6;40.6;1500.0;3.350E-1;205
10/1/2021 2:11:23 PM;Accelerating;688.8;68.1;48.9;60.6;40.8;1500.0;3.350E-1;205
10/1/2021 2:11:25 PM;Accelerating;703.6;68.8;48.9;60.6;41.0;1500.0;3.350E-1;205
10/1/2021 2:11:25 PM;Accelerating;715.0;69.3;48.9;60.6;41.2;1500.0;3.350E-1;205
10/1/2021 2:11:27 PM;Accelerating;729.6;69.4;48.9;60.6;41.4;1500.0;3.350E-1;205
```

Obrázek 3: Vstup ve formátu *RAW*.

```
210712139_211111_14.05_Results - Notepad
File Edit Format View Help
RampUpTime;RampUpValue;RampDownTime;RampDownValue;RotorTempTime;RotorTempValue
11/11/2021 2:07:12 PM;132;11/11/2021 7:53:51 PM;995;11/11/2021 7:37:14 PM;45;1
11/11/2021 8:01:08 PM;132;11/12/2021 1:47:55 AM;1004;11/12/2021 1:31:09 AM;45;2
11/12/2021 1:55:13 AM;134;11/12/2021 7:42:15 AM;1019;11/12/2021 7:25:15 AM;45;3
11/12/2021 7:49:42 AM;143;11/12/2021 1:36:20 PM;994;11/12/2021 1:19:45 PM;45;4
11/12/2021 1:43:51 PM;146;11/12/2021 7:30:45 PM;1011;11/12/2021 7:13:53 PM;45;5
11/12/2021 7:38:00 PM;131;11/13/2021 1:24:38 AM;995;11/13/2021 1:08:02 AM;45;6
11/13/2021 1:31:54 AM;132;11/13/2021 7:18:30 AM;993;11/13/2021 7:01:56 AM;44;7
11/13/2021 7:25:48 AM;134;11/13/2021 1:12:42 PM;1011;11/13/2021 12:55:49 PM;45;8
```

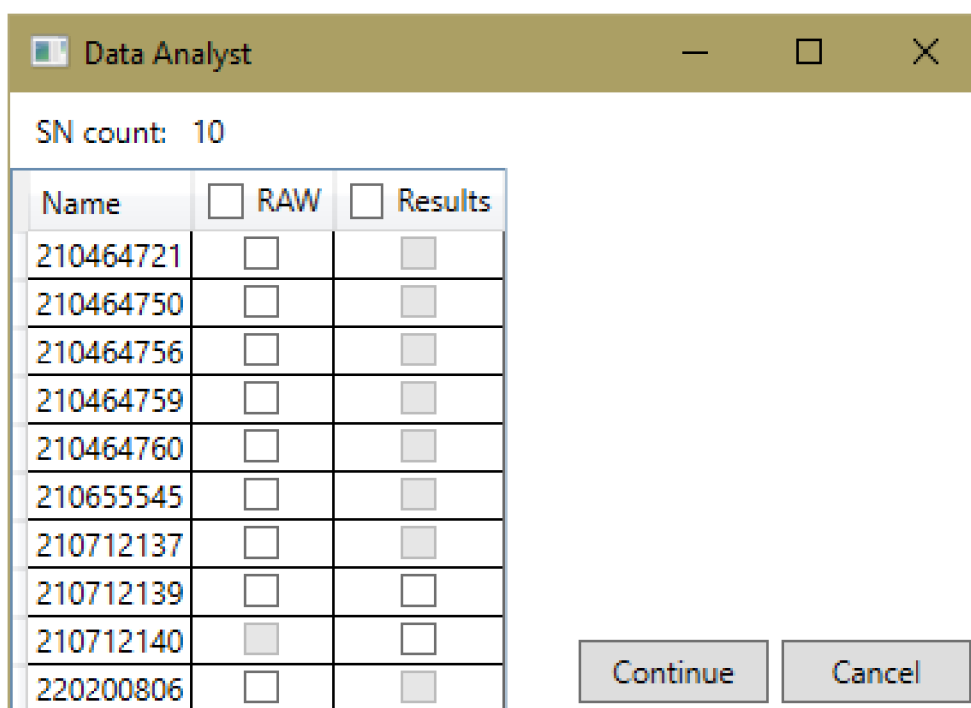
Obrázek 4: Vstup ve formátu *Results*.



Obrázek 5: Vyzvání ke zvolení vstupu.

3.2 Nastavení požadavků

Po zvolení složky se vstupními soubory má uživatel možnost vidět počet vývěv, jejichž data jsou obsažena ve vstupních souborech, jejich seznam a jaké formáty vstupních dat daná vývěva obsahuje. Jednotlivé vývěvy jsou reprezentovány jejich sériovým číslem (*SN*). Uživatel si zvolí, které vývěvy chce zpracovat a také jestli chce zpracovat pouze jeden z formátů nebo oba, pokud jsou dostupné (obrázek 6). V případě, že daný formát vstupu vývěva neobsahuje, je možnost zvolení tohoto formátu vypnuta.



Obrázek 6: Výběr vývěv a jejich vstupů.

Po výběru formátu pro jednotlivé vývěvy se zobrazí řádek pro každý ze sloupců vstupních souborů (obrázek 7). Pokud byl zvolen pouze jeden z formátů vstupních souborů, řádky pro sloupce z nevybraného formátu není možné vyplnit. Uživatel určí, které sloupce bude chtít zpracovat vyplněním jím očekávané hodnoty daného sloupce a odchylky, mimo kterou budou hodnoty považovány jako anomálie. Má také možnost spojit horní a dolní odchylku na stejnou hodnotu pomocí zaškrtnutí políčka *Link +/-* a tím vytvoří rovnoměrný pruh, ve kterém se mají hodnoty pohybovat podle očekávání. V případě vyplnění pouze jedné z hodnot odchylek a jejich nepropojení, nebude prázdná odchylka použita pro označování anomálií. Jinak řečeno, bude pruh pro očekávané hodnoty v případě vyplnění pouze horní odchylky omezen pouze shora zadanou hodnotou a jako anomálie budou označeny pouze hodnoty překračující tuto horní hranici a při vyplnění dolní hranice bude pruh očekávaných hodnot omezen pouze touto hodnotou zdola.

Data Analyst ×

Column selection:

RAW	Value:	Deviation:	
		+	-
Speed:	<input type="text"/>	<input type="text"/>	<input type="text"/>
Power:	<input type="text"/>	<input type="text"/>	<input type="text"/>
MTemp:	<input type="text"/>	<input type="text"/>	<input type="text"/>
RTemp:	<input type="text"/>	<input type="text"/>	<input type="text"/>
CTemp:	<input type="text"/>	<input type="text"/>	<input type="text"/>

Link +-

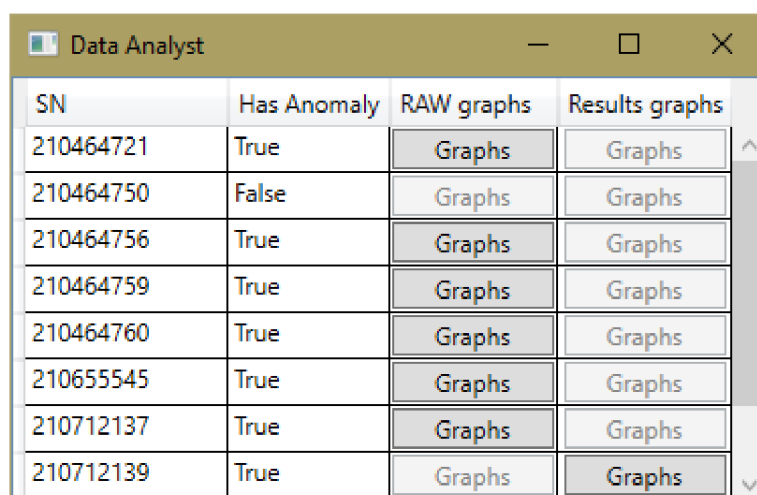
Results	Value:	Deviation:	
		+	-
RampUpValue:	<input type="text"/>	<input type="text"/>	
RampDownValue:	<input type="text"/>	<input type="text"/>	
RotorTempValue:	<input type="text"/>	<input type="text"/>	

Link +-

Obrázek 7: Vyplnění hodnot a odchylek.

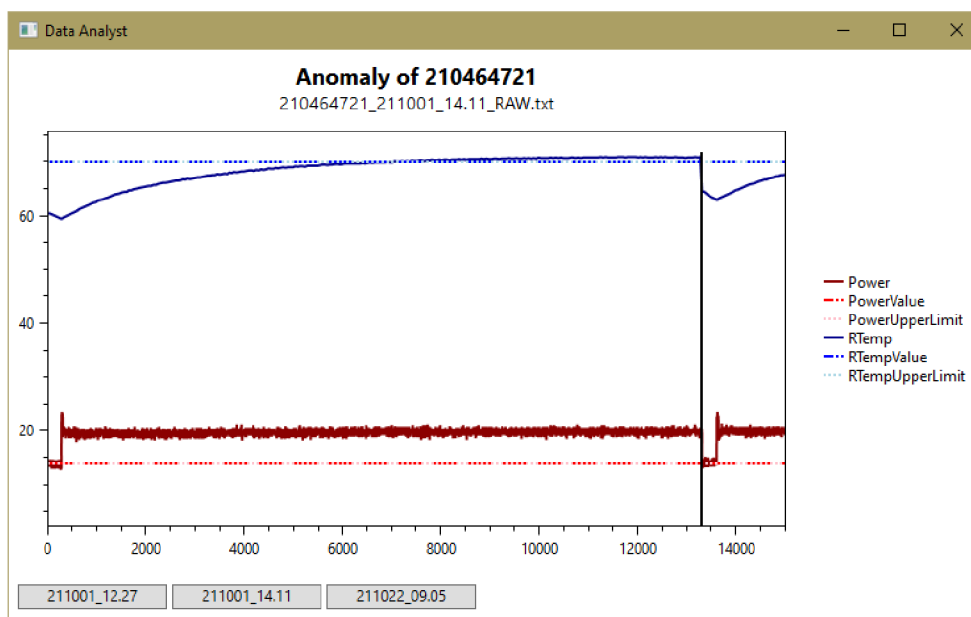
3.3 Výstupy

Po zpracování dat se uživateli zobrazí seznam obsahující sériová čísla vývěv a informaci, zda byl jejich běh anomální (obrázek 8). Vývěvy obsahující anomálii mají k dispozici možnost zobrazení grafu, ve kterém uživatel může vidět, kde hodnoty překročily jím zadaný rozsah. Uživateli je zpřístupněna možnost zobrazit graf pro ten formát vstupu, pro který byla anomálie nalezena. V grafu si uživatel může přepnout mezi vstupními soubory, ve kterých byly nalezeny anomálie (obrázek 9).



SN	Has Anomaly	RAW graphs	Results graphs
210464721	True	Graphs	Graphs
210464750	False	Graphs	Graphs
210464756	True	Graphs	Graphs
210464759	True	Graphs	Graphs
210464760	True	Graphs	Graphs
210655545	True	Graphs	Graphs
210712137	True	Graphs	Graphs
210712139	True	Graphs	Graphs

Obrázek 8: Výběr grafů k zobrazení.



Obrázek 9: Ukázka grafu.

Pro vývěvy bez anomálií je vygenerován soubor příslušející *RAW* vstupům (obrázek 10), který obsahuje průměry a mediány pro zvolené sloupce, případně maxima pro sloupce s teplotami. V případě vývěv s anomáliemi, je soubor se zpracovanými daty označen v názvu jako *Anomaly*. K anomálním vývěvám je také vygenerovaný soubor s daty cyklů, ve kterých byla nalezena anomálie. První 3 řádky pod hlavičkami reprezentují hodnoty zadané uživatelem pro jednotlivé sloupce, první řádek je zadaná hodnota, druhý řádek je horní odchylka a třetí řádek je dolní odchylka (obrázek 11). Pro *Results* vstupní data bez anomálií je vytvořený výstupní soubor, ve kterém jsou data z jednotlivých vstupních souborů spojeny do jednoho.

	A	B	C	D	E	F	G	H	I
1	Average_Speed	Median_Speed	Average_Power	Median_Power	MAX_MTemp	MAX_RTemp	MAX_CTemp	HasAnomaly	Filename
2	1499.15	1499.2	11.867073	11.9	30.5	28.3	28.1	FALSE	210464721_211104_14.55_RAW.txt
3	1498.5804	1498.5	18.125494	19.4	50.7	61.5	41.3	FALSE	210464721_211001_12.27_RAW.txt
4	1499.5101	1499.5	8.5295105	8.5	39.8	51.1	35.3	FALSE	210464721_211019_09.42_RAW.txt
5	1499.5413	1499.5	8.230667	8.2	39.6	51.1	34.9	FALSE	210464721_211021_15.10_RAW.txt
6	1499.5118	1499.5	8.585783	8.5	41.1	52	36.5	FALSE	210464721_211020_13.52_RAW.txt
7	1499.3726	1499.4	10.020358	9.9	43	52.2	36.6	FALSE	210464721_211104_15.02_RAW.txt
8	1498.4553	1498.4	19.611788	19.7	56.7	70.9	44.8	FALSE	210464721_211001_14.11_RAW.txt
9	1499.5559	1499.6	8.111024	8.1	43.3	51.6	36.2	FALSE	210464721_211019_09.42_RAW.txt
10	1499.5552	1499.6	8.141323	8.1	43.1	51.2	36.4	FALSE	210464721_211104_15.02_RAW.txt
11	1499.5406	1499.5	8.200432	8.2	40.2	49.9	34.7	FALSE	210464721_211021_15.10_RAW.txt
12	1498.4575	1498.4	19.598326	19.7	56.6	71.2	44.6	FALSE	210464721_211001_14.11_RAW.txt
13	1499.4955	1499.5	8.86343	8.9	41.6	52.1	36.3	FALSE	210464721_211104_15.02_RAW.txt
14	1499.5276	1499.5	8.161546	8.2	38.8	57.1	34.2	FALSE	210464721_211021_15.10_RAW.txt
15	1499.5652	1499.6	8.055264	8.1	41.9	50.8	35.2	FALSE	210464721_211020_13.52_RAW.txt
16	1499.5599	1499.6	8.025768	8.1	39.7	49.8	34	FALSE	210464721_211019_09.42_RAW.txt
17	1498.4735	1498.5	19.41194	19.5	56.9	71.8	44.8	FALSE	210464721_211001_14.11_RAW.txt
18	1499.5156	1499.5	8.636482	8.7	41.6	52.2	36.1	FALSE	210464721_211104_15.02_RAW.txt
19	1499.5687	1499.6	8.021846	8.1	41.6	50.8	35.2	FALSE	210464721_211020_13.52_RAW.txt
20	1499.527	1499.5	8.491445	8.5	40.9	50.9	35.1	FALSE	210464721_211104_15.02_RAW.txt
21	1498.4935	1498.5	19.206633	19.4	55.5	70.1	43.6	FALSE	210464721_211001_14.11_RAW.txt
22	1499.0896	1499.1	12.404	12.4	31.5	30	27.8	FALSE	210464721_211001_14.11_RAW.txt

Obrázek 10: Ukázka *RAW* výstupu.

1	Date_Time	PUTstat	Speed	Power	MTemp	RTemp	CTemp	SpeedRef	Vacuum	Motohour	Filename	CycleNumber
2			1	2	3	4	5					
3			5		10	1	3					
4				5	10	2	4					
5	11.04.2021 14:57	Running	1499.2	12.2	30.3	27.3	28.1	1500	0.00E+00	562	210464721_211104_14.55_RAW.txt	1
6	11.04.2021 14:57	Running	1499	12.4	30.3	27.4	28	1500	0.00E+00	562	210464721_211104_14.55_RAW.txt	1
7	11.04.2021 14:57	Running	1499	12.1	30.3	27.4	28	1500	0.00E+00	562	210464721_211104_14.55_RAW.txt	1
8	11.04.2021 14:57	Running	1499	12.5	30.3	27.4	28	1500	0.00E+00	562	210464721_211104_14.55_RAW.txt	1
9	11.04.2021 14:58	Running	1499.2	11.9	30.4	27.4	28	1500	0.00E+00	562	210464721_211104_14.55_RAW.txt	1
10	11.04.2021 14:58	Running	1499.2	11.4	30.4	27.5	28	1500	0.00E+00	562	210464721_211104_14.55_RAW.txt	1
11	11.04.2021 14:58	Running	1499.1	12.2	30.4	27.5	28	1500	0.00E+00	562	210464721_211104_14.55_RAW.txt	1
12	11.04.2021 14:58	Running	1499.1	11.3	30.4	27.6	28	1500	0.00E+00	562	210464721_211104_14.55_RAW.txt	1
13	11.04.2021 14:58	Running	1499.2	12.1	30.4	27.5	28	1500	0.00E+00	562	210464721_211104_14.55_RAW.txt	1
14	11.04.2021 14:58	Running	1499	11.3	30.4	27.5	28	1500	0.00E+00	562	210464721_211104_14.55_RAW.txt	1
15	11.04.2021 14:58	Running	1499.2	13	30.4	27.6	27.9	1500	0.00E+00	562	210464721_211104_14.55_RAW.txt	1
16	11.04.2021 14:58	Running	1499.1	12.5	30.4	27.6	27.9	1500	0.00E+00	562	210464721_211104_14.55_RAW.txt	1
17	11.04.2021 14:58	Running	1499	11.9	30.4	27.5	27.9	1500	0.00E+00	562	210464721_211104_14.55_RAW.txt	1
18	11.04.2021 14:58	Running	1498.9	12.4	30.4	27.6	27.9	1500	0.00E+00	562	210464721_211104_14.55_RAW.txt	1
19	11.04.2021 14:58	Running	1499.1	12.2	30.5	27.6	27.9	1500	0.00E+00	562	210464721_211104_14.55_RAW.txt	1
20	11.04.2021 14:58	Running	1499.2	12.1	30.5	27.7	27.9	1500	0.00E+00	562	210464721_211104_14.55_RAW.txt	1
21	11.04.2021 14:58	Running	1499.2	12.4	30.5	27.7	27.9	1500	0.00E+00	563	210464721_211104_14.55_RAW.txt	1
22	11.04.2021 14:58	Running	1499	11.6	30.5	27.7	27.9	1500	0.00E+00	563	210464721_211104_14.55_RAW.txt	1

Obrázek 11: Ukázka výstupu s anomálním cyklem.

4 Programové řešení

V této kapitole se zaměřím na zpracování samotné aplikace s bližším popisem funkcionality jednotlivých částí aplikace. Jakým způsobem je řešeno zpracování vstupů a reprezentace jejich výstupů a technologie využité pro jejich zpracování.

4.1 Použité jazyky a technologie

4.1.1 C#

C# je objektově orientovaný a typově bezpečný programovací jazyk. Typově bezpečný jazyk při kompilaci kontroluje, zda data v proměnných odpovídají zadanému typu. Objektově orientovaný jazyk je založený na konceptu „objektů“, které mohou obsahovat data a kód. Data jsou ve formě atributů a vlastností a kód ve formě metod. *C#* Umožňuje vytvářet mnoho typů zabezpečených a robustních aplikací, které běží v *.NET*. Má své kořeny v rodině jazyků *C*. Je orientovaný na komponenty a poskytuje jazykové konstrukce pro přímou podporu těchto konceptů.[11]

4.1.2 WPF

Windows Presentation Foundation je architektura uživatelského rozhraní, která je nezávislá na rozlišení a používá vektorový vykreslovací modul navržený tak, aby využíval moderní grafický hardware. WPF nabízí ucelenou sadu funkcí pro vývoj aplikací, mezi které patří XAML, ovládací prvky, datové vazby, rozložení, 2D grafika a podobně.[12]

4.1.3 .NET

.NET je vývojářská platforma pro vytváření mnoha druhů aplikací na různých platformách. Obsahuje nástroje, knihovny a jazyky, které podporují vývoj moderních, vysoce škálovatelných aplikací. Je postavena na výkonném modulu runtime (CLR), který kompiluje a spouští *.NET* programy na různých operačních systémech. CLR kompiluje kód v průběhu programování.[13, 14]

4.1.4 XAML

Extensible Application Markup Language je jazyk využívající značky. Zjednodušuje vytváření uživatelského rozhraní pro *.NET* aplikace. Deklarativními značkami XAML se mohou vytvářet viditelné prvky uživatelského rozhraní a pak definici uživatelského rozhraní oddělit od logiky aplikace pomocí souborů kódu na pozadí, které se se značkami propojují prostřednictvím definic částečných tříd.[15]

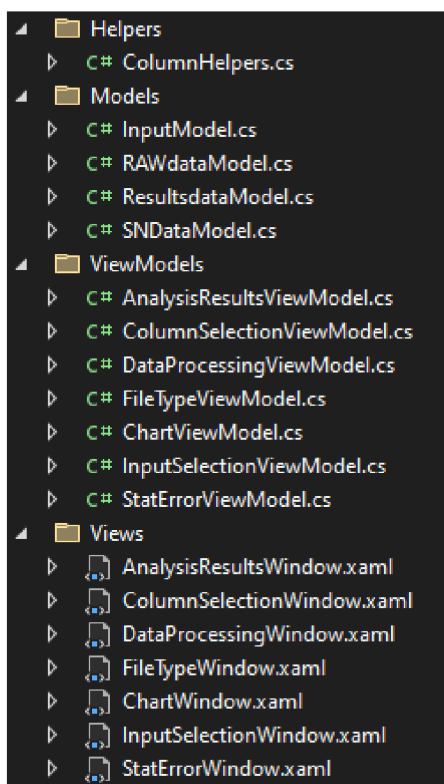
4.1.5 Použité knihovny

V práci jsem využil hned několik knihoven, které nejsou vyvíjeny přímo Microsoftem.

- **CsvHelper** je knihovna zaměřená na čtení a psaní csv souborů. Je velmi rychlá, flexibilní a jednoduchá na použití. Podporuje čtení a psaní do vlastních objektových tříd.[16]
- **MathNet.Numerics** je knihovna poskytující metody a algoritmy zaměřené na matematické výpočty.[17]
- **morelinq** je rozšířená knihovna *LINQ*, která umožňuje operace, jako je rozdělení sousedících dat do skupin podle parametru.[18]
- **OxyPlot.Wpf** je knihovna umožňující vytváření interaktivních grafů. Tento specifický balíček se zaměřuje na WPF aplikace.[19]

4.2 Architektura aplikace

Pro vývoj aplikace jsem zvolil návrhový vzor *Model-View-View Model* (MVVM) (obrázek 12). Aplikaci rozděluje na moduly, kde odděluje data, stav aplikace a uživatelské rozhraní.[20]



Obrázek 12: Složení aplikace.

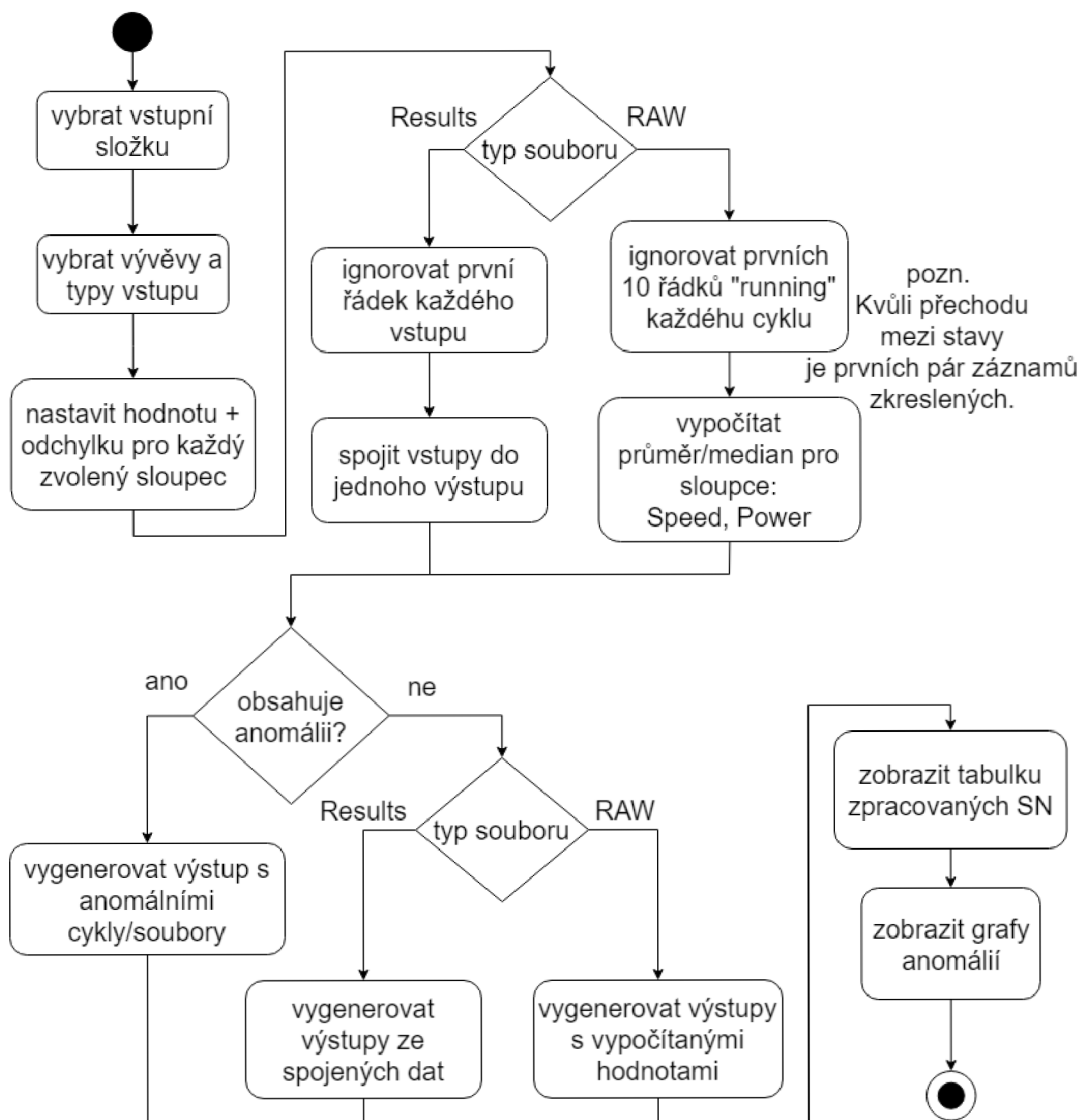
Model popisuje data, se kterými aplikace pracuje, jedná se o data přečtená ze vstupních souborů a nastavení uživatele. Model samotný neví o stavu ovládacích prvků. Mé modely definují nastavení uživatele, informace pro jednotlivé vývěvy na vstupu, určuje, jakým způsobem vypadají vstupní data *RAW* a *Results* souborů a jak mají vypadat jejich výstupní data.

View reprezentuje uživatelské rozhraní v jazyce XAML. Pomocí něj uživatel nastaví, co bude chtít se vstupními daty dělat, jaké jejich části chce zpracovat a zobrazí je po zpracování. Zobrazené informace jsou napojeny na model pomocí datových vazeb. Má aplikace obsahuje 7 views, každé pro jednotlivé okno, které se uživateli v průběhu práce s aplikací může zobrazit. Jde tedy o okno pro výběr vstupu, okno pro zvolení vývěv a formátů jejich vstupních souborů ke zpracování, okno pro zadání hodnot pro definici anomálií a výběr sloupců ke zpracování, okno s ukazatelem průběhu zpracovávání dat, okno s tabulkou zpracovaných vývěv a možnosti otevření grafů, okno pro graf a okno pro oznámení anomálií stavu běhu.

ViewModel spojuje model a view a udržuje stav aplikace. Implementuje vlastnosti a příkazy, se kterými se mohou navázat data a upozorní view na změny stavu prostřednictvím událostí oznámení o změnách. Propojení mezi view a viewModelem je pomocí vyvolávání *PropertyChanged* událostí přes *INotifyPropertyChanged* rozhraní. Každé view má svůj vlastní viewModel, který zajišťuje správné zobrazení hodnot z modelu a předání informací o uživatelském nastavení. Největšími viewModely v mé aplikaci jsou viewModel starající se o zpracování dat a viewModel, který zajišťuje vytvoření grafů.

Helpers obsahuje pomocnou třídu pro získání hodnoty ze sloupce. Je využívaný viewModely *DataProcessingViewModel* a *ChartViewModel*.

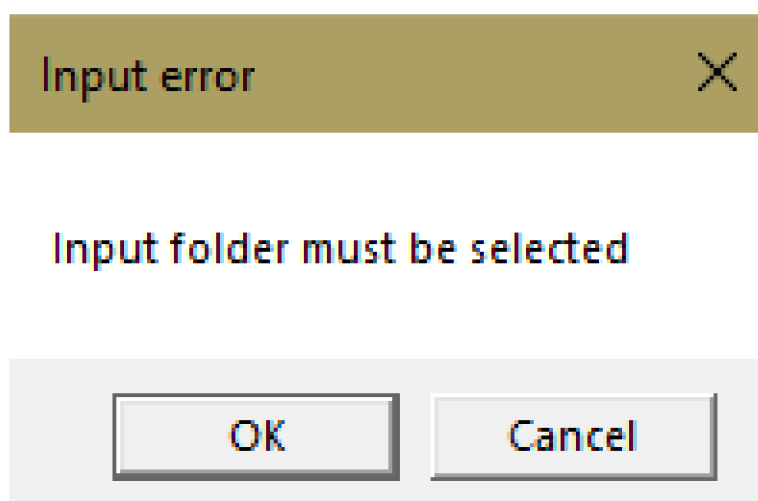
Všechny požadavky na aplikaci a její funkcionalitu se řešily přímo s firmou a v průběhu vývoje se specifikovaly. Jakým způsobem se má celá aplikace chovat se zaznamenalo před samotným programováním do diagramu (obrázek 13). Vzhled samotné aplikace součástí požadavků nebyl, takže jsem využil prostředí poskytnuté *Microsoftem*.



Obrázek 13: Diagram chodu aplikace.

4.3 Ošetření vstupů

Aplikace při jejím běhu kontroluje všechny vstupy zadané uživatelem. Jedná se o vstupní soubory, ale i hodnoty zadávané při určování rozsahu pro detekci anomálií. Při vybírání vstupní složky aplikace kontroluje, zda byla nějaká vybrána a v případě chyby nebo zavření okna pro výběr, je uživateli oznámeno, že nebyla zvolena vstupní složka a má na možnost znovu zvolit vstup nebo zrušit akci (obrázek 14). Stejně je ošetřený také výběr vstupního souboru na zobrazení grafů.



Obrázek 14: Chyba při výběru vstupu.

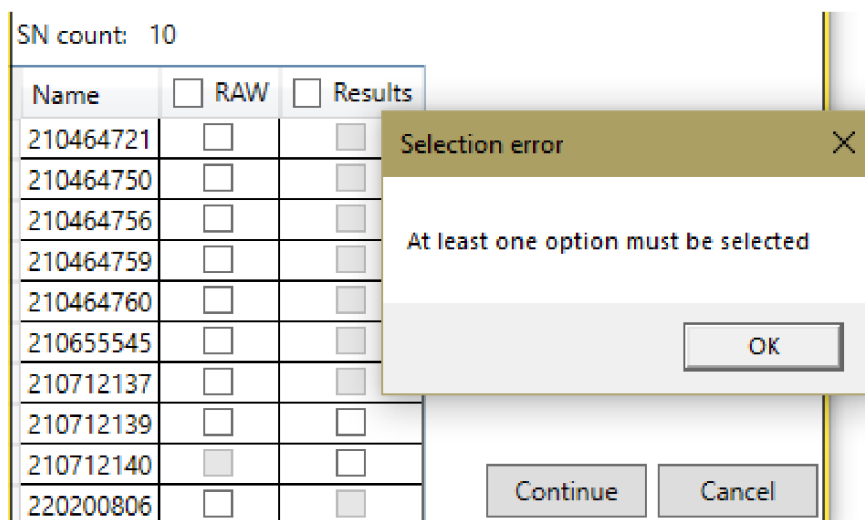
Po zvolení vstupní složky se projdou soubory ve složce. Pokud jejich názvy a formát splňují podmínku validního vstupu, tak jsou cesty k nim uloženy a z jejich názvů jsou vybrána sériová čísla a formáty souboru.

Kontrolované vlastnosti názvu souboru:

1. Prvních 9 znaků z názvu jsou číslo (sériové číslo).
2. Jsou v datovém formátu .csv nebo .txt.
3. Název obsahuje *RAW* nebo *Results*.

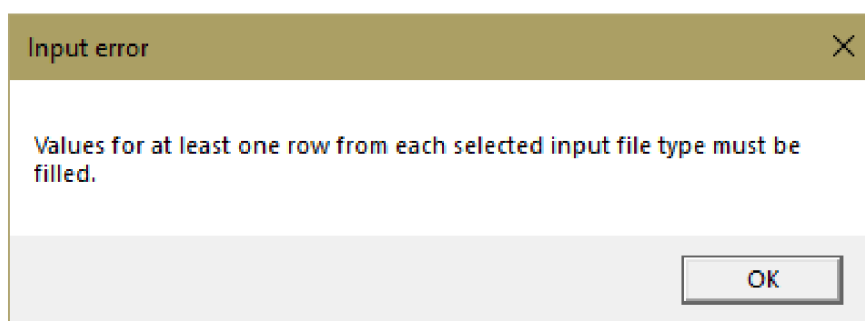
Pokud soubor nesplňuje všechny podmínky, je dále ignorován a nebude použit při zpracovávání (obrázky 15, 16, 17).

Při výběru formátu souborů pro zpracování u jednotlivých sériových čísel uživatel vybírá z dostupných možností, pokud tak neučiní, je při pokusu o pokračování upozorněn, aby označil alespoň jednu z možností (obrázek 18).



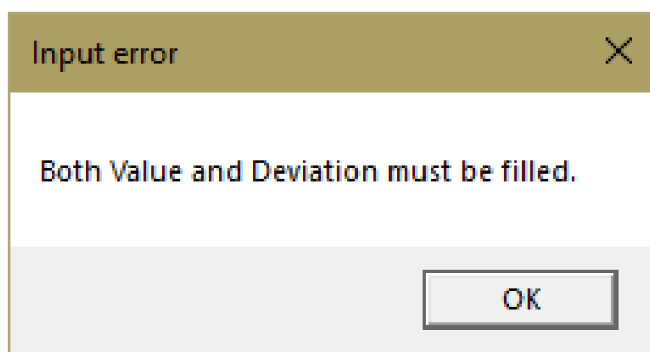
Obrázek 18: Při nezvolení vstupu pro zpracování.

Při zadávání hodnot na rozlišení anomálií jsou zaznamenány pouze číselné hodnoty. V případě zadání znaku, který není číslo, při vykliknutí z okna zmizí a není uložen. V případě, že se uživatel pokusí pokračovat bez vyplnění alespoň jedné z hodnot zvolených formátů vstupů, je na to upozorněn (obrázek 19).



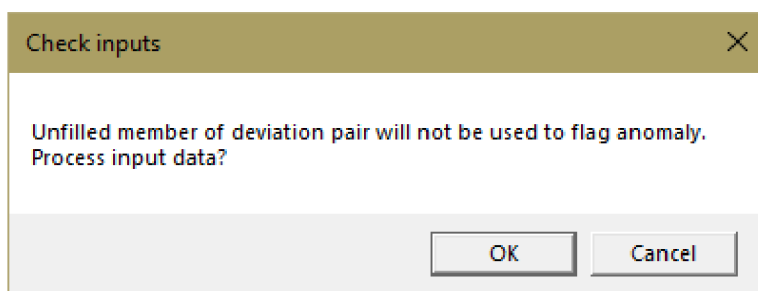
Obrázek 19: Při nezadání hodnot a odchylek.

Pokud uživatel zadá pouze očekávanou hodnotu bez odchylky nebo pouze odchylku bez očekávané hodnoty, je upozorněn, že musí být pro zvolený sloupec dat vyplněna obě pole (obrázek 20).

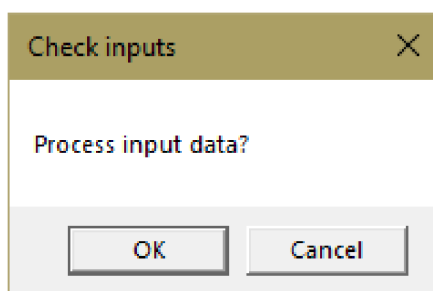


Obrázek 20: Při zadání pouze hodnoty nebo odchylky.

Uživatel má možnost propojit horní a dolní odchylku, vyplnit obě odchylky nebo pouze jednu z nich a bude obeznámen o efektu jeho volby (obrázky 21, 22). Všechny z těchto možností jsou považovány jako validní vstup.



Obrázek 21: Neprojené odchylky.



Obrázek 22: Propojené odchylky.

4.4 Re prezentace dat

Pomocí názvu souborů jsou vstupní data rozdělena na jednotlivá sériová čísla. Každé sériové číslo má přiřazenou informaci, jestliže má k dispozici *RAW*, *Results* nebo oba typy vstupů, uchovává seznam s cestami ke vstupním souborům, které mu náleží. Sériové číslo má také značku, zda pro něj byla nalezena anomálie a jestli se jedná o anomálii v *RAW* nebo *Results* vstupech. Při zvolení, jaké vstupy chce pro dané sériové číslo uživatel zpracovat, si tento model drží také tuto informaci. Pokud je pro dané sériové číslo nalezena anomálie, uloží se do něj informace o vstupním souboru s anomálií a v jakém cyklu a sloupci se objevila.

Při procházení vstupních dat samotných jsou jejich hodnoty reprezentovány modely *RAWdataModel* a *ResultsdataModel*, které obsahují definici jednotlivých sloupců a třídu pro reprezentaci jejich výstupů. Obsahují také třídu pro uživatelem zadané hodnoty a odchylky pro jednotlivé sloupce. Ty jsou využívány při znovunačtení grafů z již zpracovaných souborů.

Informace zadané uživatelem jsou uchovány v modelu *InputModel*. Mezi tyto informace patří: zvolená výstupní složka, nastavení hodnot a odchylek, počet vstupních souborů a seznam instancí jednotlivých sériových čísel.

4.5 Zpracování dat

Pro zpracovávání dat ze vstupních souborů jsem použil knihovny *CsvHelper*, *MathNet* a *MoreLinq*. Pro každé sériové číslo se asynchronně spustí funkce na zpracování vstupních dat podle držené informace, zda uživatel vybral *RAW* nebo *Results* vstupy.

Paralelně se začnou zpracovávat soubory ze seznamu vstupů přiřazených danému sériovému číslu. Není nijak omezeno, kolik souborů se bude zpracovávat najednou, ale vždy se bude najednou zpracovávat maximálně 5 vývív. Pomocí knihovny *CsvHelper* se ze souboru vytáhnou data a pomocí *morelinq* se vyfiltrují. V případě *RAW* vstupů se vyfiltrují do skupin sousedících řádků se stavem *Running*. U každé skupiny je pak přeskočených prvních 10 záznamů, kvůli nevalidním hodnotám z důvodu změny stavu. Například *Power* se bude těchto 10 záznamů stále vyrovnávat po přechodu ze stavu *Accelerating*, protože v tomto stavu vývěva potřebuje větší množství energie než ve stavu *Running*. Podle uživatelem zvolených sloupců pro zpracování se vypočítají pomocí knihovny *MathNet* průměr a medián pro sloupce *Speed* a *Power*. Pro sloupce *MTemp*, *RTemp* a *CTemp* se zjistí maximální hodnota. Na každou skupinu řádků je zavolána funkce na vyhledání anomálií.

Projdou se postupně řádky cyklu a pro každý uživatelem zvolený sloupec se zkontroluje, jestli je jeho hodnota v rozmezí, které uživatel zadal. V případě, že se ve sloupci nachází anomálie, je sériovému číslu tento sloupec označen jako anomální. U *RAW* souborů se také kontroluje, zda stav v cyklu neobsahuje hodnoty *Warning* nebo *Bellow speed ref*, které se také označí jako anomálie. Pokud nebyla pro cyklus nalezena žádná anomálie, uchovají se pouze vypočtené hodnoty. V případě *Results* souborů se nekontrolují cykly, ale jednotlivé vstupní soubory.

V případě nalezení anomálie se vytvoří ve výstupní složce dočasná složka *Temp* a do ní se pomocí *CsvHelper* vytvoří soubor s daty anomálního cyklu a informace o tomto souboru, jako je cesta k němu a číslo anomálního cyklu ze vstupního souboru. Instance sériového čísla si uchová informace o těchto dočasných souborech. Po dokončení hledání anomálií pro všechny vstupní data daného sériového čísla se zavolá vypsání výstupů do souborů. Po využití dočasných souborů na vytvoření výstupního souboru, jsou smazány pro šetření místa.

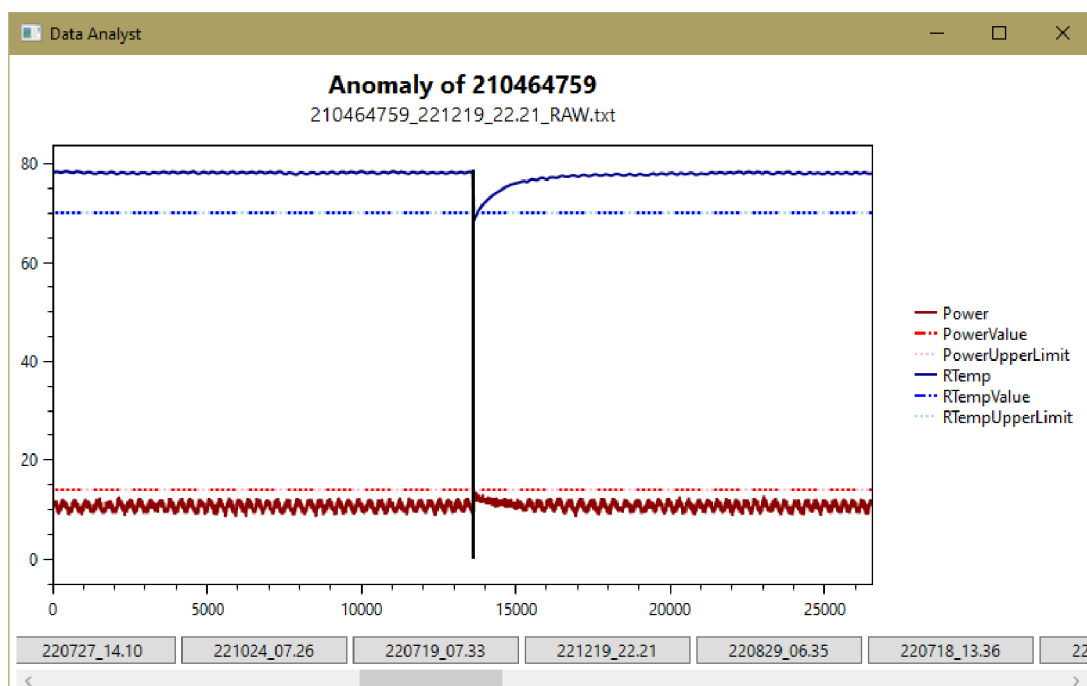
V případě, že pro sériové číslo nebyly nalezeny anomálie, se vytvoří podle zvolených vstupů složky *RAW* a *Results*, ve kterých budou uloženy výstupní soubory s vypočítanými hodnotami pro jednotlivé cykly nebo soubory. V případě anomálie se vytvoří složka *RAWAnomalyData* nebo *ResultsAnomalyData*, do kterých se vytvoří soubor s daty anomálních cyklů nebo anomálních souborů. Soubory s vypočítanými hodnotami z *RAW* vstupů pro anomální sériová čísla se vytvoří přímo do výstupní složky a budou mít v názvu *Anomaly*.

Pro anomální data se vytvoří soubor *AnomalyData*, do kterého se postupně z dočasných souborů v *Temp* složce vypíší všechny data anomálních cyklů, název vstupního souboru s anomálním cyklem a číslo tohoto cyklu ve vstupním souboru. Po projití všech sériových čísel je smazána dočasná složka *Temp*.

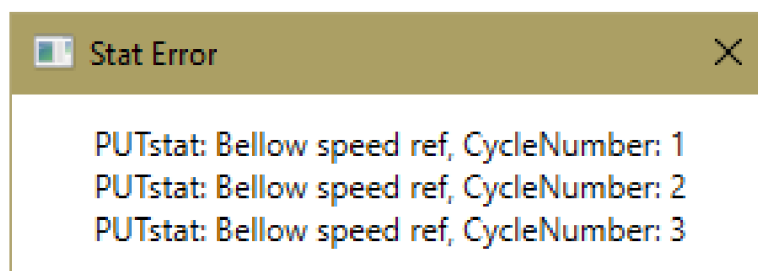
4.6 Reprezentace výstupů

Po zpracování dat je uživateli ukázáno okno se sériovými čísly, která se zpracovávala a informace, zda jim byla nalezena anomálie. Pokud ano, tak podle typu vstupu ve kterém byla, si uživatel může otevřít graf na zobrazení anomálie.

Grafy jsou generovány pomocí knihovny *OxyPlot*. Při otevření grafu se uživateli zobrazí data z prvního vstupního souboru obsahujícího anomálie. Poté má uživatel možnost nechat si vygenerovat data z ostatních vstupních souborů s anomáliemi. V grafu jsou vykreslena data ze všech zvolených sloupců ke zpracování, jaké hodnoty uživatel nastavil a odchylky. Uživatel má možnost si pomocí legendy vypnout hodnoty, které nepotřebuje vidět. Z důvodu, že hodnoty pro sloupec *Speed* se pohybují mnohem výše než z ostatních sloupců, jsou zmenšeny 100násobně a je jim přiřazena jednotka reprezentující změnu měřítka. Funkcionalita knihovny *OxyPlot* umožňuje uživateli si přiblížit části grafu a zobrazit hodnotu jednotlivých bodů. V případě, že je anomálie nalezena ve více cyklech jednoho vstupního souboru, jsou na grafu tyto cykly odděleny svislými černými čárami (obrázek 23). V případě, že se v daném vstupním souboru našla anomálie ve stavu běhu, je uživatel upozorněn při vykreslení grafu daného vstupního souboru (obrázek 24).



Obrázek 23: Graf s více anomálními cykly.



Obrázek 24: Anomálie ve stavu běhu.

Vygenerované soubory s daty pro anomální cykly jsou ve stejném formátu jako na vstupu, s rozdílem, že je přidána informace o názvu vstupního souboru a číse cyklu s anomálií.

Pro *RAW* zpracované vstupy sériových čísel s anomálií i bez anomálie je vygenerovaný soubor s vypočítanými hodnotami pro sloupce zvolené ke zpracování pro každý cyklus, název vstupního souboru obsahující daný cyklus a informace, zda v daném cyklu byla nalezena anomálie. Tento soubor se liší mezi sériovými čísly s anomálií a bez anomálie pouze názvem a umístěním. Pokud pro dané sériové číslo byla nalezena anomálie, tento soubor bude přímo ve výstupní složce s názvem ve formátu *Anomaly_RAWSeriovéČíslo*. Pro sériová čísla bez anomálií bude tento soubor ve složce *RAW* a v názvu má pouze *RAWSeriovéČíslo*. Pro zpracované *Results* vstupní soubory bez anomálie jsou data spojena do jednoho

výstupního souboru a jediná změna je přidání sloupců s označením, zda obsahují anomálie a název jejich vstupního souboru.

5 Poznatky

V této poslední kapitole se zaměřím na porovnání mé aplikace s předchozím způsobem zpracovávání souborů z testování vývĚv. Jakým způsobem jsem zefektivnil analýzu těchto dat a hlavní rozdíly mezi mým řešením a řešením pomocí *PowerQuery*.

5.1 Kvalita řešení

Pro porovnání kvality mého řešení použiji jako protiklad předchozí nejlepší existující řešení a to je *PowerQuery*. Vzhledem k tomu, že v případě *PowerQuery* je řešení aplikované pouze na *RAW* vstupní soubory a ty jsou také řádově větší než *Results*, bude porovnání stavěno na zpracování právě jich. Tento formát vstupních souborů navíc drží více informací potřebných pro analýzu běhu vývĚvy.

Jeden z faktorů kvality řešení je jednoduchost používání. V případě *PowerQuery* může uživatel nastavovat odchylky a hodnoty pro vyhledání anomálií pouze po otevření kódu query a změněním hodnot v něm (obrázek 25). Pro tyto úpravy musí mít uživatel alespoň základní znalosti o tom, jak *PowerQuery* funguje. Také musí při změně vstupu upravit cestu k němu přímo v kódu.

```
let
    Source = Folder.Files("C:\Users\Jakub\Desktop\Práce\RAW-uprava\RAW\210664758\mesic"),
    #Filtered Hidden Files! = Table.SelectRows(Source, each {Attributes}[Hidden] < true),
    #Invoke Custom Function! = Table.AddColumn(#Filtered Hidden Files!, "Transform File", each #"Transform File"({Content})),
    #Renamed Columns! = Table.RenameColumns(#Invoke Custom Function!, {"Name", "Source.Name"}),
    #Removed Other Columns! = Table.SelectColumns(#Renamed Columns!, {"Source.Name", "Transform File"}),
    #Expanded Table Columns! = Table.ExpandTableColumn(#Removed Other Columns!, "Transform File", Table.ColumnNames(#"Transform File"("#Sample File"))),
    #Removed Columns! = Table.RemoveColumns(#Expanded Table Columns!, {"Date.Time", "Speed", "SpeedRef", "Vacuum", "Not@hours"}),
    #Renamed Columns! = Table.RenameColumns(#Removed Columns!, {"Source.Name", "File"}),
    #Replaced Value! = Table.ReplaceValue(#Renamed Columns!, " ", "", Replacer.ReplaceText, {"Power", "MTemp", "RTemp", "CTemp"}),
    #Changed Type! = Table.TransformColumnTypes(#Replaced Value!, {"Power", type number}, {"MTemp", type number}, {"RTemp", type number}, {"CTemp", type number}),
    #Added Anomaly! = Table.AddColumn(#Changed Type!, "Anomaly", each if {PUStat} = "Below speed ref" or {PUStat} = "Warning" then "Y" else if {Power} < 5 or {Power} > 15 then "I" else "N"),
    #Replaced Errors! = Table.ReplaceValue(#Added Anomaly!, each {PUStat}, each if {PUStat} = "Below speed ref" or {PUStat} = "Warning" then "Warning" else {PUStat}, Replacer.ReplaceText, {PUStat}),
    #Grouped Rows! = Table.Group(#Replaced Errors!, {"Cycle", each _, type table [File:text, PUStat:text, Power:nullable number, MTemp:nullable number, RTemp:nullable number, CTemp:nullable number]},
    #Filtered Rows! = Table.SelectRows(#Grouped Rows!, each {PUStat} = "Running"),
    #Remove first! = Table.AddColumn(#Filtered Rows!, "Custom", each Table.RemoveFirstN({Cycle}, 10)),
    #Removed Columns! = Table.RemoveColumns(#Remove first!, {"PUStat", "Cycle"}),
    #Added Index! = Table.AddColumn(#Removed Columns!, "Index", 1, Int64.Type),
    #Expanded Custom! = Table.ExpandTableColumn(#Added Index!, "Custom", {"File", "Power", "MTemp", "RTemp", "CTemp", "Anomaly"}, {"File", "Power", "MTemp", "RTemp", "CTemp", "Anomaly"}),
    #Finalised! = Table.Group(#Expanded Custom!, {"Index"}, {"Average_Power", each List.Average({Power}), type number}, {"Max_MTemp", each List.Max({MTemp}), type number}, {"Max_RTemp", each List.Max({RTemp}), type number}, {"Max_CTemp", each List.Max({CTemp}), type number}),
in
    #Finalised
```

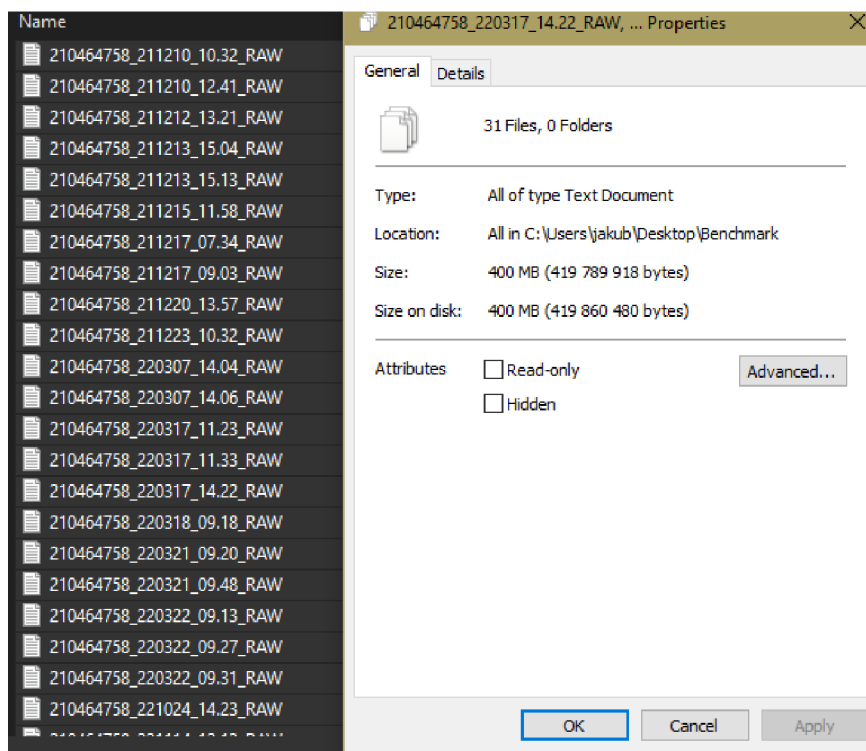
Obrázek 25: Ukázka query pro anomálie v *Power* sloupci.

V mém programu je zadávání hodnot řešeno vizuálně a uživatel je upozorněn při nesprávném vyplnění. Výstupem query je pak tabulka v *Excelu*, která oproti mému řešení neobsahuje informace o jiných sloupcích než *Power*, *Mtemp*, *Rtemp* a *Ctemp* (obrázek 26). Uživatel také může vytvořit graf pouze z těchto vypočítaných hodnot, a pro zobrazení běhu u anomálního cyklu si musí dohledat vstupní soubor a ručně vytvořit graf v *Excelu*. Pokud daný vstupní soubor obsahuje příliš mnoho řádků, *Excel* není schopný tolik řádků uchovat a uživatel musí odhadem vytáhnout pouze část řádků ze vstupního souboru, kde očekává, že se anomálie může nacházet.

Index	Average Power	Max_Mtemp	Max_Rtemp	Max_Ctemp	File	Anomaly
1	11.62352941	32.4	28.4	29.6	210464758_211203_13.48_RAW.txt	FALSE
2	9.654455081	46.3	58	40.5	210464758_211203_13.56_RAW.txt	FALSE
3	8.996200764	46.1	59.1	40.5	210464758_211203_13.56_RAW.txt	FALSE
4	8.734161947	45.7	60.8	40.2	210464758_211203_13.56_RAW.txt	FALSE
5	8.337955399	45.9	62.3	40.7	210464758_211203_13.56_RAW.txt	FALSE
6	8.558117457	46.1	65.2	40.7	210464758_211203_13.56_RAW.txt	FALSE
7	8.505127262	45.4	65.5	40.2	210464758_211203_13.56_RAW.txt	FALSE
8	8.474919165	45.1	65.9	39.9	210464758_211203_13.56_RAW.txt	FALSE
9	8.467276602	45	66	39.5	210464758_211203_13.56_RAW.txt	FALSE
10	8.45646929	45.2	66.2	39.8	210464758_211203_13.56_RAW.txt	FALSE
11	8.448717572	43	65.1	37.5	210464758_211203_13.56_RAW.txt	FALSE
12	8.431325301	42.4	64.9	37.3	210464758_211203_13.56_RAW.txt	FALSE
13	8.443396921	43.3	65.4	38	210464758_211203_13.56_RAW.txt	FALSE
14	8.460019147	43.8	65.9	38.6	210464758_211203_13.56_RAW.txt	FALSE
15	8.469762657	42.8	64.9	37.1	210464758_211203_13.56_RAW.txt	FALSE
16	8.486851974	42	64.7	36.7	210464758_211203_13.56_RAW.txt	FALSE
17	8.506822354	42.8	65.3	37.6	210464758_211203_13.56_RAW.txt	FALSE
18	8.499985275	43.8	66.2	38.5	210464758_211203_13.56_RAW.txt	FALSE
19	8.481179755	43.6	65.3	37.2	210464758_211203_13.56_RAW.txt	FALSE
20	8.525425969	42.4	65.4	37.1	210464758_211203_13.56_RAW.txt	FALSE
21	8.54192	42.7	65	37.3	210464758_211203_13.56_RAW.txt	FALSE
22	13.93923749	49.6	60.5	41	210464758_211210_10.32_RAW.txt	TRUE
23	12.98030403	52.6	66.1	43.2	210464758_211210_12.41_RAW.txt	TRUE

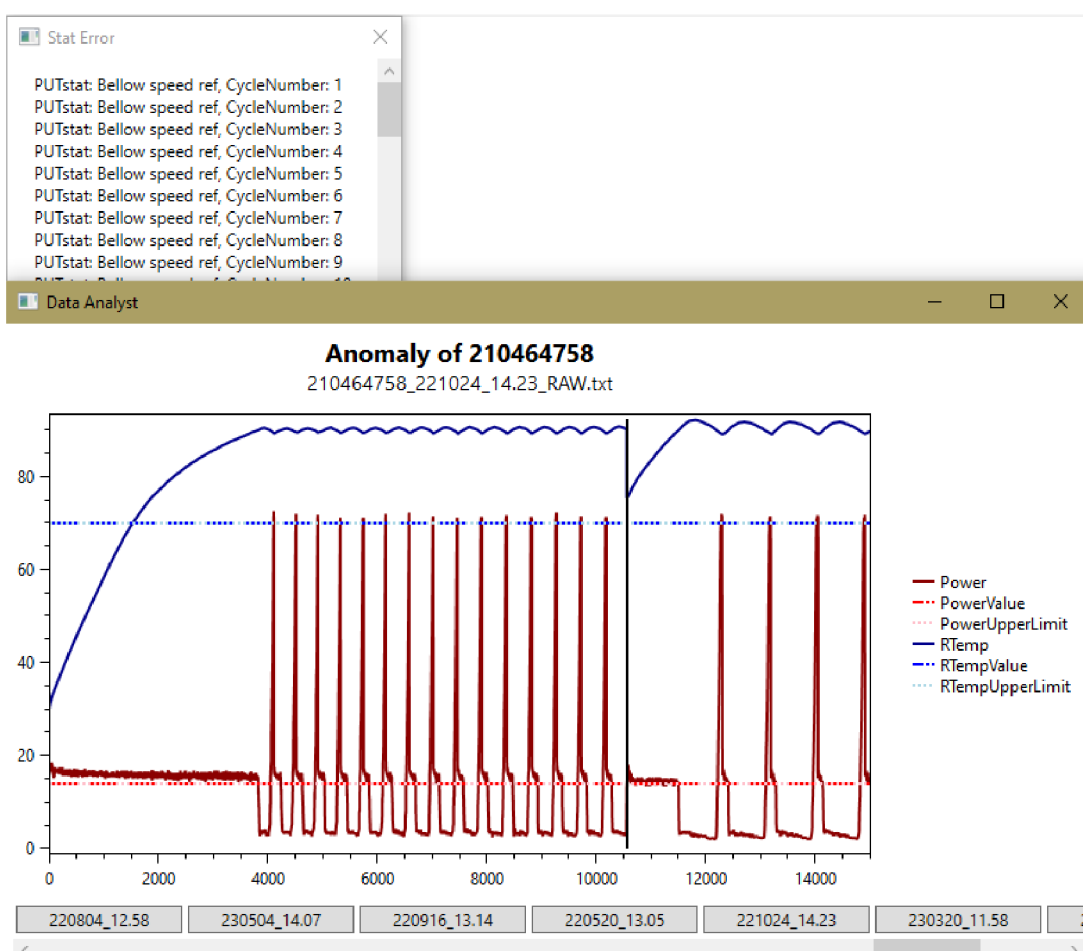
Obrázek 26: Výstup z query.

Dalším faktorem pro určení kvality řešení, je rychlost zpracování vstupních souborů. Jako testovací vzorek jsem zvolil vstup o velikosti 400 MB (obrázek 27).



Obrázek 27: Vstup na test rychlosti.

Mé řešení zvládlo tato vstupní data zpracovat během třiceti vteřin, zatímco query ani po dvou hodinách běhu a plného využití operační paměti data nezpracovalo. Při rozdělení vstupních dat na dvě poloviny, byla první část o velikosti 183MB zpracována za šestnáct minut a druhá část o velikosti 217MB se zpracovávala šestnáct a půl minuty. Zobrazení anomálních cyklů je v mém řešení k dispozici hned po dokončení zpracování, zatímco v případě query je stále potřeba ručně vyhledat anomálie ve vstupních souborech. Také v případě anomálie stavu běhu, jako je *Warning* nebo *Bellow speed ref* nejsou ve výstupu z query nijak speciálně označeny, zatímco v mém řešení je uživatel obeznámen o jejich typu a lokaci (obrázek 28). Při kontrolování 2 sloupců na vstupu o velikosti 30 GB byly data zpracovány mým řešením za 33 minut.



Obrázek 28: Ukázka anomálií ve stavu běhu.

5.2 Budoucí vývoj

Hlavní způsob, jakým by se dalo mé řešení zlepšovat, je rychlost zpracování dat. Přesto, že momentálně zvládá mnohem větší vstupy než původní řešení, stále existují způsoby, jak zpracování urychlit. Velkým faktorem, který zpomaluje zpracování, jsou operace čtení a psaní souborů.

Další zlepšení by mohlo být v rychlosti načítání grafů. V případě, že bude na vstupu velký soubor, jehož běh by byl celý označený jako anomální, a uživatel zadal na zpracování všechny sloupce, graf by musel najednou vykreslit desítky až stovky tisíc bodů. Navigace v tomto grafu by s použitím aktuální knihovny byla poté téměř nemožná. Momentálně je tento problém řešený pomocí přiblížení na menší sekci grafu, čímž se omezí potřeba vykreslení velkého množství bodů.

Kromě zlepšení rychlosti se nabízí vylepšení univerzálnosti řešení. Umožnění definice oddělovače v csv vstupech, možnost zpracování jiných formátů vstupních dat nebo specifikace formátu výstupních dat. Mé řešení se zaměřuje pouze na požadavky firmy, která měla potřebu zpracovávat data v pevném formátu s jasnou představou formátu výstupu.

Zvážil jsem také možnost napojení na databázi a ukládání zpracovaných vstupů do ní. Ve firmě, která tento program bude aplikovat, je nejbližší věc k databázi, kterou používají, úložiště *SharePoint*. Vygenerované soubory by šly nahrávat na SharePoint v případě, že by aplikace běžela na zařízení připojeném do sítě, ale z důvodu, že část zařízení ve firmě vůbec nejsou připojeny do sítě, by byla tato možnost nadbytečná. Bylo by důležité také ošetřit bezpečný přenosu těchto dat, jelikož se jedná o interní informace o výrobcích.

Závěr

Výstupem práce je okenní aplikace, která je schopná zpracovávat velké desítky GB vstupních dat ve formátu csv se sloupci oddělenými středníkem. To v existujícím řešení ve firmě nebylo možné udělat efektivním způsobem.

Uživatel vybere složku se vstupními soubory, určí, které vývěvy chce zpracovat a jaký formát vstupů, definuje očekávané hodnoty a odchylku pro sloupce vstupů, které chce zpracovat a určí výstupní složku. To vše bez potřeby znalostí programovacích jazyků. Po zadání všech potřebných hodnot, to znamená pouze výběr složky, vývěv na zpracování a vyplnění rozmezí hodnot pro určení anomálií, stačí pouze počkat na vyplnění ukazatele průběhu.

Po zpracování má uživatel možnost vidět, které ze zpracovávaných vývěv obsahovaly anomálie a v kterém typu vstupu byly. Má možnost zobrazit grafy na analýzu těchto anomálií. Také má k dispozici výstupní soubory se zpracovanými daty a v případě anomálií také soubory se vstupními daty obsahující anomálie.

Zatímco předchozí řešení zvládlo vstup o velikosti 400 MB zpracovat, po manuálním rozdělení tohoto vstupu a nutnosti spuštění query na každý z nich, za více než 30 minut, můj program byl za stejnou dobu po jednom počátečním nastavení hodnot schopen zpracovat za stejnou dobu 30 GB vstupních dat.

V případě chyby zpracování u query se uživatel nikdy nedostal ani k částečně zpracovaným výsledkům a v mém programu má i při výpadku k dispozici alespoň částečně zpracované výstupy.

Při vytváření této práce jsem využíval znalosti z práce s vakuovými vývěvami. Aktivně komunikoval s firmou a upřesňoval jejich požadavky. Práce bude aktivně využívána pro analyzování běhů testů vakuových vývěv a mnohonásobně sníží nutnost trávit čas touto analýzou.

Conclusions

The output of the work is a windowed application capable of processing large datasets, tens of GBs, in CSV format with semicolon-separated columns. This was not possible in the existing solution at the company in an efficient manner.

The user selects a folder with input files, specifies which processes and inputs to process, defines expected values and deviations for the input columns to be processed, and determines the output folder. All of this without requiring knowledge of programming languages. After entering all the necessary values, which only involve selecting the input folder, the process to be analyzed, and filling in the range of values to determine anomalies, the user only needs to wait for the progress indicator to complete.

Upon completion, the user has the option to see which processed processes contain anomalies and in which type of input they were found. Graphs for analyzing these anomalies are also available. Processed data output files and, if applicable, input data files containing anomalies are available to the user.

While the previous solution managed to process a 400 MB input in over 30 minutes, with the manual splitting of inputs and the need to run queries on each of them, my program was able to process 30 GB of input data in the same amount of time after a single initial setup.

In the case of query processing errors, the user never obtained partially processed *Results* in the previous solution, whereas my program provides at least partially processed outputs even in the event of a failure.

During the development of this work, I leveraged knowledge of working with vacuum pumps. I actively communicated with the company to refine their requirements. This work will be actively used for analyzing test runs of turbo-molecular pumps, significantly reducing the need to spend time on this analysis.

A Obsah elektronických dat

Zdrojové kódy, text práce a instrukce jsou v elektronickém systému katedry informatiky.

doc/

Adresář s textem práce ve formátu PDF, a soubor .zip obsahující zdrojový text teoretické části a použité obrázky.

src/

Adresář obsahující soubory aplikace, zdrojové kódy jsou rozdělené podle architektury MVVM.

install/

Adresář obsahující soubory k instalaci programu a jeho spuštění.

test_data.zip

Archiv obsahující soubory pro testování aplikace.

- Vývěva s *RAW* daty s anomáliemi
- Vývěva s *RAW* daty bez anomálie a *Results* daty s anomáliemi
- Vývěva s *Results* daty bez anomálie

hodnoceni_Edwards

Ohodnocení mého programu firmou *Edwards*.

README.txt

Textový soubor obsahující instrukce pro instalaci aplikace a její použití.

Literatura

- [1] *Edwards Vacuum trhy. : Analytické přístroje* [online]. [cit. 2023-7-11]. Dostupný z: <https://www.edwardsvacuum.com/cs-cz/our-markets/analytical-instrumentsf>.
- [2] *Edwards Vacuum trhy. : Chemický a potravinářský průmysl* [online]. [cit. 2023-7-11]. Dostupný z: <https://www.edwardsvacuum.com/cs-cz/our-markets/chemical-processing-industries>.
- [3] *Edwards Vacuum trhy. : Výzkum a vývoj* [online]. [cit. 2023-7-11]. Dostupný z: <https://www.edwardsvacuum.com/cs-cz/our-markets/research-and-development>.
- [4] *Edwards Vacuum trhy. : Výroba energie* [online]. [cit. 2023-7-11]. Dostupný z: <https://www.edwardsvacuum.com/cs-cz/our-markets/energy>.
- [5] *Edwards Vacuum trhy. : Polovodiče* [online]. [cit. 2023-7-11]. Dostupný z: <https://www.edwardsvacuum.com/cs-cz/our-markets/semiconductor-and-electronics>.
- [6] *Edwards Vacuum trhy. : Displej* [online]. [cit. 2023-7-11]. Dostupný z: <https://www.edwardsvacuum.com/cs-cz/our-markets/display>.
- [7] *Edwards Vacuum produkty. : Mechanické turbomolekulární vývěvy* [online]. [cit. 2023-7-11]. Dostupný z: <https://www.edwardsvacuum.com/cs-cz/our-products/turbomolecular-pumps/next-mechanical-turbomolecular-pumps>.
- [8] *Edwards Vacuum specifikace. : Technická specifikace* [online]. [cit. 2023-7-11]. Dostupný z: https://digitalbrochure.edwardsvacuum.com/nEXT-TURBOMOLECULAR-PUMPS/en/#/acf/table/block_613752ef10b1a.
- [9] *Edwards produkty. : Typy vývěv od Edwards* [online]. [cit. 2023-7-24]. Dostupný z: <https://www.edwardsvacuum.com/en-uk/our-products>.
- [10] *Edwards vývěva. : Turbomolekulární vývěva* [online]. [cit. 2023-7-24]. Dostupný z: <https://www.edwardsvacuum.com/en-uk/our-products/turbomolecular-pumps>.
- [11] *Learn Microsoft. : Prohlídka jazyka C#* [online]. [cit. 2023-7-11]. Dostupný z: <https://learn.microsoft.com/cs-cz/dotnet/csharp/tour-of-csharp/>.
- [12] *Learn Microsoft. : Průvodce pro desktop (WPF .NET)* [online]. [cit. 2023-7-11]. Dostupný z: <https://learn.microsoft.com/cs-cz/dotnet/desktop/wpf/overview/?view=netdesktop-7.0>.
- [13] *Learn Microsoft. : Co je .NET?* [online]. [cit. 2023-7-11]. Dostupný z: <https://learn.microsoft.com/cs-cz/dotnet/core/introduction>.
- [14] *Amazon. : What is .NET?* [online]. [cit. 2023-7-11]. Dostupný z: <https://aws.amazon.com/what-is/net/>.

- [15] *Learn Microsoft. : Přehled XAML* [online]. [cit. 2023-7-11]. Dostupný z: <https://learn.microsoft.com/cs-cz/dotnet/desktop/wpf/xaml/?view=netdesktop-7.0>.
- [16] *Csvhelper. : .NET library for reading and writing CSV files* [online]. [cit. 2023-7-24]. Dostupný z: <https://joshclose.github.io/CsvHelper/>.
- [17] *Math.NET Numerics.* [online]. [cit. 2023-7-24]. Dostupný z: <https://numerics.mathdotnet.com>.
- [18] *MoreLINQ. : Extensions to LINQ to Objects* [online]. [cit. 2023-7-24]. Dostupný z: <https://morelinq.github.io>.
- [19] *OxyPlot. : Plotting library for .NET* [online]. [cit. 2023-7-24]. Dostupný z: <https://oxyplot.github.io>.
- [20] *Learn Microsoft. : Model-View-ViewModel (MVVM)* [online]. [cit. 2023-7-11]. Dostupný z: <https://learn.microsoft.com/cs-cz/dotnet/architecture/maui/mvvm>.