Signature-based Virus Detection for Windows Operating Systems Using Java and PostgreSQL

Eliyas Sala

I. INTRODUCTION

Did you know that Windows is the most used operating system? In fact, 76.12% of computer consumers used Windows operating system in 2021 [4]. Since it has existed for such a long period of time compared to other operating systems, hackers and cyber terrorists have found loopholes to manipulate computer systems running this operating system. They have created different malwares, like viruses, that are often undetectable by antivirus softwares [2]. The data above motivates individuals, like me, who seek business opportunities by developing a software to secure this system. Although different antivirus softwares exist, most of them are ineffective or lack the consumers' trust. As part of the requirements for this project in my Operating Systems course, this research briefly discusses how these programs work and their tested performances. Then I propose my software-based solution with different development tools.

II. RELATED WORK

Sukwong et al. examines the most used AV softwares and the algorithmic implementation behind them. He/she first provides the distinction between Signature-based detection vs. Behavior-based detection [5]. Both are ways to detect virus and every AV product doesn't necessarily use the same method to do that. In Signature-based detection, AV product scans a file and assigns a unique identification to that file. For example, it could use hashing algorithms like MD5 to assign value. Then it evaluates based on patterns of that hashing by comparing it to a remote database containing viral characteristics. If a match exists, then that file is indeed virus infected. Behavior-based detection, another common implementation, analyzes the behaviors of that specific file instead of accuracy matching. It is important to note that there are many more other implementations that different AV softwares use, but we will focus on these two.

Sukwong et al. continues to discuss and points out the ineffectiveness of AV softwares. Based on an experiment conducted to test the responsiveness of selected AV softwares, their activities were monitored when intentionally given infected files. The softwares in the experiment were the

following: Avast, Kaspersky, McAfee, Norton, Symantec, and Trend Micro. Avast performed the best by detecting 62.15% of malware. However, it couldn't fully detect all of them. This clearly shows we shouldn't completely rely on our AV softwares to keep our windows computers secure.



Figure 1 shows comparisons of different AV softwares

As you can see above, 42.83 percent of the malware remained undetected on first time. Among these undetected virus-infected files, 53.83 percent produced one or more malicious other files. Additionally, we found that 23.49 percent of these undetected malware files with malicious child files had one or more child files detected within zero days, but 64.36 percent weren't detected. This data tells us that there are limitations to these AV softwares.

III. IMPLEMENTATION

To solve the problem of enhancing virus detection capability on Windows operating systems, it is important to analyze the different types of viruses that exist and learn their behaviors. Based on the statistics provided on the global market share of desktop operating systems, Windows has consistently dominated the market for years in staggering amounts. Other well-known operating systems, like macOS, are far from reaching this success. However, this much of consumer usage has created vulnerability for Windows operating systems [4]. Unlike most antivirus software, my application performs detection-only service and does not have a removal feature. And it builds upon a signature-based detection system.



Figure 2 shows my software

Given a directory path to scan for viruses, my software will first use a hashing algorithm to compute the MD5 hashing for every file in that directory. Then it matches strings collections of stored MD5 data in a PostgreSQL database containing large data of the existing viruses and their MD5 classification signatures [3]. After cross-referencing the MD5 generated from scanning the files against our data of known viruses' MD5, my application decides whether is it safe or virus infected. If infected, a pop-up message should appear before the users warning them.

IV.

WORK

Overall goal

Deciding between Microsoft Visual Studio to develop a C/C# based program and Java-based desktop application. Since I have the most experience with Java, I chose to use Eclipse IDE to write my code and develop my program.



Figure 3 shows java libraries and data structures I used

I decided which programming langauge to use based on feasibility. I choose Java (my most proficient langauge) and I designed the user interface using Swing API. I created event listeners for the components as well. I imported Java's security class to use MessageDigest class to access it and generate MD5 hashing by scanning the files. The data structures I used include File object to store files as arrays and an arraylist to store MD5 strings. I also used a Scanner object to scan the files and store them as strings for manipulation.

C	8	https	://virus	sshare	.com/hasl	nfiles/\ ×	+				
####################################	\leftarrow	\rightarrow	C		virussh	are.com/	nashfile	es/Viru	usShar	e_003	71.md5
Malware sample MD5 list for # VirusShare_00371.zip # http://VirusShare.com # Twitter: @VXShare.com # cd3664f12f8b565967941c13c0e76 d540c580da351801f25e19865c25d8 ie48a0cf7c9e8788531376854343797 d5294119e20d390b418e0fcbdde633 d5fae946d1620b33248e3f3eadc5178 f40db6e9649b9cc82cde28d3ef8d2c fbf997f3d411c9f2c52f57eef8ea5e cf557ca454fecbb3ec886de8661d7 is98c7b7cb922d9d54a7274c5825bf3 V41bcf94da6fb36db2cdda9f44c39 reaa38abc05aa31583493ad29dbe9e V1850f3022fdd542c810db14030300e bcc95519a58ab07fa2d25feb36891d5 l866b7cd91fb20dd793b82187079d V4453adbe210ef84e0045e2a11fa1da V10c0da45ca9af2b8cd622dadb155 i609cdf619c972dde1d93778b8c817f b198e8f20b67fb2384ae881f967 id6a02c0c5cc7a3f493a62841e41705 i607e1f8b6add372c0e4978a3777 i56799ac3559842e3d5472c99bdeed Ved4021d8ca13bf12aa28a0e9abcd17 ia5b28f6702d8a60cf53796d24219f 1697ab81bcaa6f2c48abc0608a339 7c9e38ce51dd7bc81c32242847d						*****					
VirusShare 00371.zip # http://VirusShare.com # Twitter: @VXShare # ###################################	t Mal	lware	e samr	le M	D5 list	for #					
http://Yinushare.com # Twitter:@VXShare.dots Twitter:@VXShare.m Twitte	ŧ Vir	nusSk	are 6	00371	zip	#					
Twitter: # ####################################	# htt	tp://	Virus	Shar	e.com	#					
####################################	ŧ Twi	itter	·: @V)	Shar	e	#					
cd366e4f12f8b566967941c13c0e76 d540c580da351801f25a19865c25d8 ie48a0cf7c9e8788531376854343797 d6294119e200390b418e0fcbddde33 d65ae946d1620b33248e3f3ea4c5178 f400b6e9649b9c82cde28d3ef8d2c f400b6e9649b9c82cde28d3ef8d2c f5575a452b4fecbb3ec8866de8661d7 ie398c7b7cb922d9d5a7274c5825bf3 W11bcf94ad6fb36db22cdd9f44c39 fea38babc05aa31583493ad29dbe9e W18861251be9c2d90d5a31df3529b 20f33022fdd542c810ddb14030300e hcc95519a58ab07fa2d25feb36831d5 Bie6b7cd196b20dd793b82187079d V4453adbe210ef84e0045e2a1811da 'd10c0dad5c3a972b8c482478 id10c0dad5c3a972b8c482176 hJ98e8f220b67f3b2384ae881f967 id6a02c0c5cc7a3f493a62841e41705 i6079c4f619c372c0e47983777 i56799ac3559842e3d54472c99bdeed led40a21d8ca1a5bfae8fa4a8b616c l5C8af933a6145f12aa28a0e9abcd17 ia5b28f6702d8a60cf53796d24219f 1697ab81bcaa6f2c48abc0868a339 '7e9e58ce51db15ca8f2c48abc0868a339 '7e9e58ce51db15ca8f2c48abc0868a399	#####	*####	*#####	****	*****	#######					
d540c580da35180tf25c19865c25d8 ie48a0cf7c9e878851376854343797 i6294119c20d390b418e0fcbddde633 i6fae946d1620b33248e3f3ea4c5178 if40db6e9649b9cc82cde28d3ef8d2c ifbf997f3d411c9f2c52f57eef8ea5e i745565905127.0cf9521616566aa7 ic557ca454afecbb3ec886de0861d7 is98c7b7cb922d9d547274c5825bf3 i411bcf94ad6fb36db22cdda9f44c39 ieaa3b8abc05aa3183433ad29dbe9e i418861251be9c2d90dd5a31dfa529b i20f33022fdd542c31dfa529b i20f33022fdd542c31dfa529b i866b7cd1f6b206d7993b82187079d i4463adbe210ef84e0045e2a21811da i410c0da4f5c39af28cd62431dfa529b i609cdf619c972dde1d93778b8c817f b19880f220b67f3b2384aea681f967 id6a02c0c5cc7a3f493a62841e41705 i56799ac3559842a30547ae2029bdeed ied40a21d8ca1a5bfaee8fa488b616c i56399a3559842a30547ae209bdeed ied40a2148ca1a5bfaee8fa48b616c i5c8af933a6145f12a28a60e3a39 i609cdf619c924da40475a93624219f i609cd8b1caa6f2c48abc06088a339 ieg98c6702d8a60cf53796d24219f i607ab3b1caa6f2c48abc03608a339 ieg98c8c51dbd7b681c3224208471d i67fa5d6e0008816a960b681c4105bc	5ccd3	366e4	1f12f8	36566	967941c	13c0e76					
ie48a0cf7c9e87885137685434797 i629411920d390b418e0fcbdde633 i6fae946d1620b33248e3f3ea4c5178 if40db6e9649b9c62cd28d3ef8d2c if5bf997f3d411c9f2c52f57eef8ea5e i745565905127c0cf05521615566aa7 ic557ca454facbb3ec8a86de8061d7 ia98c7b7cb922d9d54a7274c5825bf3 i411bcf94ad6fb36db2cdda9f44c39 reaa38abc05aa31583493ad29dbe9e i418661251be9c2d90d5a31dfa529b '20f33022fdd542c810db14030300e bcc5551aa58ab07fa2d25feb36831d5 i8e6b7cd91f0b206d7993b82187079d i4463adbe210ef84e0045e2a21811da 'd10c0da45ca9a72b8cd52adabb95 i609cdf619c972dde1d93778b8c817f ib19e80f20b67f3b2384aea681f967 id6a02c0c5cc7a3f493a62841e41705 i67c1f8k6ad4372c0e49278a3777 i56799ac3559842e3d54472c99bdeed bed40a21d8ca1a5bf2ea8fa4a8b616c i5C8af933a6145f12aa28a0e9abcd17 ia5bb28f6702d8a60cf53796d24219f i1697ab81bcaa6f2c48abc03608a339 '7e9e58ce51db1bc881c3224208471d i67fa5d6e0008816e3960bcd	fcd54	10c58	30da35	51801	f25e198	65c25d8					
<pre>6294119e20d390b418e0fcbdde633 65da946d1620b33248e3f3ea4c5178 f400b6e9649b9c482cd28d3ef8d2c f409b6e9649b9c482cd28d3ef8d2c f4745565905127c0cf05521616566aa7 ic557ca454afecbb3ec886de0861d7 ia98c7b7cb922d9d54a7274c5825bf3 4411bcf94ad6fb36db22cdda9f44c39 eaa3b8abc05aa3158449ad29dbe9e 4418861251be9c2d90dd5a31dfa529b 20f33022fd542c3810ddb14030300e icc95519a58ab07fa2d25feb36891d5 1866b7cd1f6b206d7993b82187079d 44463adbe210ef84e0045e2a21811da 'd10c0da45ca9a72b8cd62adabb95 :609cdf619c972dde1d93778b8c817f 199e80f230b67fb328dea681f967 :d6a02c0c5cc7a3f493a62841e41705 :767c1f8b46ad3727c0e49e78a3777 i56799ac3559842e3054712c99bded ted40a21d8ca1a5bfaee8fa488b61c i5c8af93a6145f12a28a0e3bcd17 ia5b28f6702d8a60cf53796d24219f :1697ab81caa6f2c48abc0368a339 '7e9e58ce51dbd7b681c3224208471d 167fa5d6e000881c6a966bcd1105bc</pre>	f5e48	3aØc1	F7c9e8	37885	3137685	4343797					
:6fae946d1620b3248e3f3ea4c5178 :f40d566949b3c282cd22d3ef8d2c :fbf997f3d411c9f2c52f57eef8ea5e :745565965127c0cf05521615666aa7 :c557ca454afecbb3ec8a86de0861d7 :a98c7b7cb922d9d54a7274c5825bf3 V411bcf94ad6fb36db22cda9f44c39 'eaa3b8abc05aa31583493ad29dbe9e V418661251be9c2d90d5a31dfa529b :20f33022fdd542c810ddb14030300e :cc95519a58ab0f7ad25feb36891d5 :8e6b7cd91f0b206d7993b82187079d V4463adbe210ef84e0045e2a21811da 'd10c0da4f5ca9af2b8cd62adabb95 :609cdf619c972dde1d93778b8c817f :b19e80f220b67f3b2384ae881f967 :id6a02c0c5cc7a3f493a62841e41706 :f07c1f8b6add372c0e40978a3777 :56799ac3559842e3d54472c99bdeed !e440a21d8ca1a5bfae8fa4a8b616c :5C8af933a6145f12aa28a0e9abcd17 :a5bb28f6702d8a60cf53796d24219f :1697ab81bcaa6f2c48abc03608a339 '7e9e58ce51db7b681c3224208471d 167fa5d6e0008816e3960bcd1105bc	6e629	94119	e20d3	390Ь4	18e0fcb	ddde633					
f40db6e9649b9c482cd28d3ef8d2c fbf997f3d411c9f2c5f55perf8ea5e i745565995127c0cf05521616566aa7 ic557ca454afecbb3ec8866de8661d7 ia98c7b7cb922d9d54a7274c5825bf3 4411bcf94ad6fb36db22cdda9f44c39 eaa3b8abc05aa31583493ad29dbe9e 4418861251be9c2d90dd5a31dfa529b 20f33022fdd542c810ddb14030300e bcc95519a58ab07fa2d25feb36891d5 88c6b7cd1f6b206d7993882187079d 4463adbe210ef84e0045e2a21811da 'd10c0da45ca9af2b8cd62adabb95 :609cdf619c972dde1d93778b8c817f .b19e80f220b67f3b2384aea81f967 :id6a02c0c5cc7a3f493a62841e41705 :767c1f8b46a4d3727c0e49e78a3777 :56799ac3559842e3054712c99bdeed ted40a21d8ca1a5bfaee8fa488b16c :5c8af93a6145f12a28a0e9abcd17 ia5bb28f6702d8a60cf53796d24219f :1697ab81caa6f2c48abc03608a339 '7e9e58ce51db7bc81c3224208471d 167fa5d6e000881c6a966bdc1105bc	7c6fa	ae946	5d1620	эьзз2	48e3f3e	a4c5178					
fbf997f3d411c9f2c52f57eef8ea5e f745565965127c0cf65521616566aa7 ic557ca454afecb3ec8a86de0861d7 ia98c7b7cb922d9d54a7274c5825bf3 i411bcf94ad6fb36db22cdda9f44c39 reaa58abc05aa31583493ad29dbe9e i418661251be9c2d90d5a314fa529b i20453022fdd542c810ddb14030300e icc55519a538ab07fa2d25feb36891d5 i8e6b7cd91f0b206d7993b82187079d i463adbe210ef84e045e2a21811da id10c0da4f5ca9af2b8cd62adadbb55 i609cdf619c972dde1d93778b8c817f i928e320b67f3b2384aec881f967 id6a02c0c5cc7a3f493a62841e41705 i672r164b6ad43727c0e49e78a3777 i56799ac3559842e3d54472c99bdeed ied40a2148ca1a5bfae8fa4a8b616c i528af933a6145f12aa28a0e9abcd17 ia5b28fc702d8a60cf53796d24219f i697ab81bcaa6f2c48abc0808a339 '7e9e58ce51dbd7b681c3224208471d i67fa5d6e0000816a960bcd	9ff40	odb6e	9649b	9008	2cde28d	3ef8d2c					
1745565905127.0cf05521616566aa7 ic557ca454facbb3ec8a86de8861d7 ia98c7b7cb922d9d54a7274c5825bf3 Vaaa3b8abc05aa31583493ad29dbe9e Vaa3b8abc05aa31583493ad29dbe9e Vaa3b8abc05aa31583493ad29dbe9e Vaf33022fdd542c810ddb14030300e bcc95519a58ab07fa2d25feb36891d5 NBc6b7cd916b206d7993B82187079d V4463adbe210ef84e0045e2a21811da V410c0da4f5ca9af2b8cd62adabb95 i609cdf619c972dde1d93778b8c817f b19e80f220b67fb2384aea681f967 id6a02c0c5cc7a3f493a62841e41705 i6767c1f8b46ad43727c0e49e78a3777 i56799ac3559842e3d54472c99bdead Ved4021d8ca1a5bfaee8fa4a8b616c I5C8af933a6145f12aa28a0e9abcd17 ia5bb28f6702d8a60cf53796d24219f 1697ab81bcaa6f2c48abc03608a339 '7e9e58ce51dbd7b681c3224208471d I67fa5df6e0008816a966bcd	01fb†	F9971	F3d411	Lc9f2	c52f57e	ef8ea5e					
<pre>ic557ca454afecbb3ec8a66de0861d7 is08c7b7cb922d954a7274c5825bf3 i411bcf94ad6fb36db22cdda9f44c39 reaa5b8abc05aa31583493ad29dbe9e i418861251be9c2d90d5a31dfa529b 20f33022fdd542c810ddb14030300e icc95519a58ab07fa2d25feb36891d5 i8e6b7cd91f0b206d7993b82187079d i4463adbe210ef84e0045e2a21811da 'd10c0da4f5ca9af2b8cd62adadb95 i609cdf619c972dde1093778b8c817f ib19e80f220b67f3b2384aea681f967 id6a02c0c5cc7a3f493a62841e417065 i767c1f8b46add3727c0e49e78a3777 i56799ac3559842e3d54472c99bdeed ied40a21d8ca1a5bfae8fa488b616c i5c8af933a6145f12aa28a0e9abcd17 ia5bb28f6702d8a60cf53796d24219f i607ab81bcaa6f2c48abc03608a339 '7e9e58ce51dbd7b681c32242088471d i67fa5d6e000881c6g960bded1105bc</pre>	94745	55659	905127	/cØcf	0552161	6566aa7					
iag8c7b7cb922d9d54a7274c5825bf3 Villbcf94ad6fb36db22cda9f44c39 Yeaa3b8abc05aa31583493ad29db99e Vil861251be9c2d90d5a31dfa529b Y20f33022fdd542c810ddb14030300e bcc55519a58ab07fa2d25feb36891d5 N8c6b7cd91f6b206d7993b82187079d Vil0c0da4f5ca9af2b8cd62adabb95 6099cdf619c972dde1d93778b8c817f hJ98e8f220b67f3b2384aea681f967 Vid6c0ad4f5ca9af2b2384aea681f967 Vid6c0ad472c09472b384c881f967 Vid6c0ad45ca9af27c0e49e78a3777 Vi67c1f8b46ad4372c0e49e78a3777 Vi56799ac3559842e3d54472c99bdeed VedA021d8ca1a5bfaee8f4488b616c Vi5C8af933a6145f12aa28a0e9abcd17 Vi567af67b12aa8ae6f396d24219f Vi697ab81caa6f2c48abc036068a339 '7e9e58ce51dbd7b681c3224208471d I67fa5d6e0008816e966bdcd1105bc	95c55	57ca4	154afe	ecbb3	ec8a86d	e0861d7					
H411bcf94ad6fb36db22cdda9f44c39 Peaa3b8abc05aa31583493ad29dbe9e H418bc05aa31583493ad29dbe9e H418861251be9c2d90dd5a31dfa529b Pcomposition Pcomposition Pcomposition H418bc1251be9c2d90dd5a31dfa529b Pcomposition Pcomposition Pcomposition Pcomposition Pcomposition H453adbe210ef84e0045e2a1811da H463adbe210ef84e0045e2a1811da H463adbe210ef84e0045e2a1811da H463adbe210ef84e0045e2a1811da H463adbe210ef84e0045e2a1811da H463adbe205c7ca73f493a62841e41705 H5709ac3559842e3d54472c99bdeed H4040a2148ca1455fae88fa48b616c H568933a6145f12aa28a0e9abcd17 H30528f6702d8a66cf53796d24219f H50528f6702d8a66cf53796608a339 7e9e58ce51dbd7b681c3224208471d H67fa5d6e0000816c49660b641105bc	d5a98	3c7b7	7cb922	2d9d5	4a7274c	5825bf3					
<pre>'eaa3b8abc05aa31583493ad29dbe9e 'ta18861251be9c2d90d5a314fa529b '20f33022fdd542c810ddb14030300e bcc95519a58ab07fa2d25feb36891d5 88e6b7cd91f0b206d7993b82187079d 14463adbe210ef84e0045e2a21811da 'd10c0dad45ca9af2b8cd62adabb95 .609cdf619c972dde1d93778b8c817f .b198e8f220b67f3b2384aea881f967 .id6a02c0c5cc7a3f493a62841e41705 .f6799ac3559842e3d54472c99bdeed !ed40a21d8ca1a5bfae8fa4a8b616c .f5C8af933a6145f12aa28a0e9abcd17 .a5bb28f6702d8a60cf53796d24219f .1697ab81bcaa6f2c48abc03608a339 '7e9e58ce51dbd7b681c32242088471d .67fa5d6e0008816e966bded1105bc</pre>	c9411	lbcfg	94ad61	Fb36d	b22cdda	9f44c39					
H418861251be9c2d90dd5a31dfa529b Y20f33022fdd542c810ddb14030300e >cc95519a58ab07fa2d25feb36891d5 Bs6b7c401f0b20647993b82187079d H4463adbe210ef84e0045e2a21811da Yd10c8da4f5ca9af28cd62adabb95 :609cdf619c972dde1d93778b8c817f :b19e80f220b67f3b2384aea681f967 :d6a02c0c5cc7a3f493a62841e41705 :767c1f8b46add3727c0e49e78a3777 :56799ac3559842e3165472c99bdeed !ed40a21d8ca1a5bfaee8fa4a8b616c :5c8af933a6145f12aa28a0e9abcd17 :a5b28f6702d8a60cf53796d24219f :1697ab81caa6f2c48abc03608a339 '7e9e58ce51dbd7bc81c3224208471d :67fa5d6e0000816ed960bded1105bc	a7eaa	a3b8a	abc05a	aa315	83493ad	29dbe9e					
20f33022fdd542c810ddb14030300e 2c95519a58ab07fa2d25feb36891d5 18e6b7cd91f0b206d7993b82187079d 1463adbe210ef84e0045e2a21811da 1d10c0da4f5ca9af2b8cd62adadbb95 1609cdf619c972dde1d93778b8c817f 1b19e80f220b67f3b2384aea881f967 1d6a02c0c5cc7a3f493a62841e41705 1767c1f8b46add3727c0e49e78a3777 156799ac3559842e3d54472c99bdeed 1ed40a2148ca1a5bfae8fa4a8b616c 15C8af933a6145f12aa28a0e9abcd17 1a5bb28f6702d8a60cf53796d24219f 1697ab81bcaa6f2c48abc03608a339 7e9e58ce51dbd7b681c3224208471d 167fa5dfe00008816e3968bded1105bc	94418	38612	251be9	e2d9	0dd5a31	dfa529b					
<pre>bcc95519a58ab07fa2d25feb36891d5 88c6b7cd91f6b206d7993b82187079d V4463adbe210ef84e0045e2a21817079d V4463adbe210ef84e0045e2a21811da d10c0dadf5ca9af2b8cd62adabb95 609cdf619c972dde1d93778b8c817f b19880f220b67f3b384aea681f967 66a02c0c5cc7a3f493a62841e41705 767c1fBb46add3727c0e49e78a3777 156799ac3559842e3d5472c99bdeed Ved40a21d8ca1a5bfaee8fa4a8b616c 15c8af933a6145f12aa28a0e9abcd17 ia5bb28f6702d8a60cf53796d24219f 1697ab81bcaa6f2c48abc03608a339 7e99e58ce51dbd7b681c3224208471d 167fa5d6e0008816a966bbcd</pre>	c7201	F3302	22fdd	542c8	10ddb14	030300e					
88e6b7cd91f0b206d7993b82187079d W463adbe210ef84e0045ea221811da Vd10c0da4f5ca9af2b8cd62adadb95 5609cdf619c972dde1093778b8c817f bJ9e80f220b67f3b2384aea681f967 Vd6a02c0c5cc7a3f493a62841e41706 Yd7c1f846add3727c0e49e78a3777 Y5679ac3559842e3d54472c99bdeed Ved40a2148ca1a5bfae8f4a8b616c I5c8af933a6145f12aa28a0e9abcd17 Ia5bb28f6702d8a60cf53796d24219f 1697ab81bcaa6f2c48abc03608a339 '7e9e58ce51dbd7b681c3224208471d I67fa5d6e0000816ed960bded1105bc	dØccs	95519	a58ab	07fa	2d25feb	36891d5					
V4453adbe210ef84e0045e2a21811da V410c0da4f5ca9af28e3de2adabb95 :609cdf619c972dde1d93778b8c817f :b19e80f220b67f3b2384aea681f967 :d6a02c0c5cc7a3f493a62841e41705 :767c1f8b46a4d3727c0e49e78a3777 :56799ac3559842e3d54472c99bdeed Ved40a21d8ca1a5bfaee8fa4a8b616c :5c8af933a6145f12aa28a0e9abcd17 :a5bb28f6702d8a60cf53796d24219f :1697ab81bcaa6f2c48abc03608a339 '7e9e58ce51dbd7b681c3224208471d 167fa5d6e0000816e3060b6d1105bc	888e6	5b7co	191fØb	206d	7993b82	187079d					
<pre>'d10coda4f5ca9af2b8cd62adadb95 609cdf619c972dde1d93778b8c817f b19e80f220b67f3b2384aea681f967 id6a02c0c5cc7a3f493a62841e41705 5767c1f8b46add3727c0e49e78a3777 556799ac3559842e3d54472c99bdeed ied40a2148ca1a5bfaee8fa4a8b616c 15c8af933a6145f12aa28a0e9abcd17 ia5bb28f6702d8a60cf53796d24219f '1697ab81bcaa6f2c48abc03608a339 '7e9e58ce51dbd7b681c3224208471d 167fa5d6e0000816ed960bded1105bc</pre>	f9446	53adb	e210e	ef84e	0045e2a	21811da					
609cdf619c972dde1d93778b8c817f bl9e80f220b67f3b2384aea681f967 id6a02c0c5cc7a3f493a62841e41705 y67c1f8b46ad43727c0e49e78a3777 y56799ac3559842e3d54472c99bdeed yed40a21d8ca1a5bfae8fa4a8b616c y5c8af933a6145f12aa28a0e9abcd17 ia5bb28f6702d8a60cf53796d24219f 1697ab81bcaa6f2c48abc03608a339 '7e9e58ce51dbd7b681c3224208471d l67fa5d6e0000816e30960bdc41105bc	77d10	0c0da	a4f5ca	a9af2	b8cd62a	dadbb95					
b19e80f220b67f3b238Aea681f967 d6a02c0c5c7a3f493a62841e41705 b767c1f8b46a4d3727c0e49e78a3777 b56799ac3559842e3d54472c99bdeed bed40a2148ca1a5bfaee8fa488b616c b5c8af933a6145f12aa28a0e9abcd17 da5b28f6702d8a60cf53796d24219f d597ab81bcaa6f2c48abc03608a339 7e9e58ce51dbd7b681c3224208471d l67fa5d6e0000816ed960bded1105bc	6c609	edfe	519c97	72dde	1d93778	b8c817f					
id6a02c0c5cc7a3f493a62841e41705 y767c1f846ad43727c0e49278a3777 y567c9ac3559842e3d54472c99bdeed yed40a2148ca1a5bfaee8fa4a8b616c y5c8af933a6145f12aa28a0e9abcd17 ia5bb28f6702d8a60cf53796d24219f 1697ab81bcaa6f2c48abc03608a339 '7e9e58ce51dbd7b681c3224208471d 167fa5d6e0000816e30960bded1105bc	31619	e801	F220b6	57f3b	2384aea	681 f 967					
767c1f8b46ad3727c0e49e78a3777 156799ac3559842e3d5472c99bdeed 1ed40a21d8ca1a5bfaee8fa4a8b616c 15c8af933a6145f12aa28a0e9abcd17 1a5bb28f6702d8a60cf53796d24219f 1697ab81bcaa6f2c48abc03608a339 7e9e58ce51dbd7b681c3224208471d 167fa5d6e0000816e30960bded1105bc	95d6a	a02c6	c5cc7	7a3f4	93a6284	1e41705					
156799ac3559842e3d54472c99bdeed bed40a2148ca1a5bfae8fa4a8b616c 15c8af933a6145f12aa28a0e9abcd17 la5bb28f6702d8a60cf53796d24219f 1697ab81bcaa6f2c48abc03608a339 7e9e58ce51dbd7b681c3224208471d 167fa5d6e0000816ed960bded1105bc	eb767	7c1f8	3b46a4	1d372	7c0e49e	78a3777					
<pre>>ed40a21d8ca1a5bfaee8fa4a8b616c i5c8af933a6145f12aa28a0e9abcd17 ia5bb28f6702d8a60cf53796d24219f i1697ab81bcaa6f2c48abc03608a339 '7e9e58ce51dbd7b681c3224208471d i67fa5d6e0000816ed960bded1105bc</pre>	13567	799ac	:35598	342e3	d54472c	99bdeed					
15c8af933a6145f12aa28a0e9abcd17 1a5b228f6702d8a60cf53796d24219f 1697ab81bcaa6f2c48abc03608a339 7e9e58ce51dbd7b681c3224208471d 167fa5d6e0000816c4960bded1105bc	d9ed4	10a21	Ld8ca1	La5bf	aee8fa4	a8b616c					
ia5bb28f6702d8a60cf53796d24219f 1697ab8lbcaa6f2c48abc03608a339 7e99e58ce51dbd7b681c3224208471d 167fa5d6e0000816ed960bded1105bc	9d5c8	3af93	33a614	15f12	aa28a0e	9abcd17					
1697ab81bcaa6f2c48abc03608a339 /7e9e58ce51dbd7b681c3224208471d 167fa5d6e0000816ed960bded1105bc	43a5ł	ob281	F67020	18a60	cf53796	d24219f					
77e9e58ce51dbd7b681c3224208471d 167fa5d6e0000816ed960bded1105bc	1f169	₹7ab	31bcaa	a6f2c	48abc03	608a339					
167fa5d6e0000816ed960bded1105bc	177e9	€58¢	e51db	od7b6	81c3224	208471d					
	2d671	Fa5de	5e0000	9816e	d960bde	d1105bc					
10bdbe926acde29e93317186b68dcfa	oa0bo	lbe92	26acde	29e9	3317186	b68dcfa					

Figure 4 shows the data source I used for identifying viruses

I used a real-world dataset from VirusShare.com. First, I stored the MD5 data in text files and then, I used Heroku to create a PostgreSQL database and added database driver to eclipse. Then I wrote a class for a database connection and data insertion. Every time my application runs, it should either return "clean" or "virus detected". After selecting a directory, users click "scan" button to run the start the process for detection. The backbone of my application rests on the MD5 hashing function and Checksum functions. I created a byte array to read data in chunks. Note that "checksum" function as a parameter of previous function's data. I also incorporated exception handling for better code quality.

V. CONCLUSION

This research-based project explores the different virus detection systems for Windows operating system. Then it discusses a signature-based scanning system with its implementation. Although my database only contains 10,000 rows of data due to resource limitations, the application is fully scalable. The UI needs improvements however the functionality is intact. My work does not claim superiority compared to the other AV solutions; however, it provides a different approach to tackle the problem. It will continue to undergo refinement until my desired goals are met and optimal solutions are achieved.

- Al-Asli, Mohammed et al. "Review of Signature-based Techniques in Antivirus Products." (2019).
- [2] Garba, Faisal A. et al. "Evaluating the State of the Art Antivirus Evasion Tools on Windows and Android Platform." (2019).
- [3] Sahoo, Abhaya, et al. "Signature based Malware Detection for Unstructured Data in Hadoop." (2014).
- [4] "Desktop operating system market share worldwide," StatCounter Global Stats. [Online]. Available: https://gs.statcounter.com/os-marketshare/desktop/worldwide. [Accessed: 05-Sep-2021].
- [5] O. Sukwong and H. S. Kim, "Commercial Antivirus Software Effectiveness: An Empirical Study," IEEE Computer Society, pp. 63-70, 2011.